目　录

第1章　离散时间信号的时域分析

1.1　导言

数字信号处理是处理一种称为输入信号的离散时间信号,以产生另一种称为输出信号的离散时间信号,这种输出信号具有更多人们所需要的特性。在某些应用中,需要使用特定的数字信号处理算法来提取原信号中的一些重要特性。通过观察系统对给定输入信号的输出信号,还可以研究离散时间系统的特性。因此,首先学会用 MATLAB 在时域中产生一些基本的离散时间信号,并对这些信号进行一些基本的运算,是很重要的,这也正是第 1 章练习的主要目的。第二个目的是学会使用基本的 MATLAB 命令,并将它们应用到简单的数字信号处理问题中。

1.2　入门

随本书一起赠送的磁盘提供了本书中的所有 MATLAB 程序,还提供了在 PC 和 Macintosh PowerPC 下都适用的未完成的报告。磁盘中按照章节顺序,还特别提供了前 10 个练习中 PC 和 Macintosh 版本的 MATLAB M 文件和 Word 版本的报告,这些文件按照章节存放在不同的子目录下。完成每一个实验练习例子后,读者可以在指定的位置完成有关该练习的一些问题。

PC 下的练习安装向导

按照下面的步骤,可将包含程序和报告的文件夹复制到装有 Windows 95/98 系统的计算机硬盘中:

1. 插入软盘。
2. 双击桌面上的**我的电脑**图标,打开**我的电脑**窗口。
3. 双击相关图标,打开软驱窗口。
4. 双击相关图标,打开将安装该文件夹的目的硬盘驱动器窗口。
5. 在软驱窗口中,将标记为 **PC** 的文件夹拖到目的硬盘驱动器的目录中。

Macintosh PowerPC 下的练习安装指导

按照下面的步骤,可将包含程序和报告的文件夹复制到装有 Mac OS 或更高版本的 PowerPC 的硬盘中:

1. 插入软盘。
2. 双击桌面上显示的图标,打开硬盘驱动器窗口。
3. 双击相关图标,打开软驱的窗口。
4. 在软驱窗口中,将标记为 **MAC** 的文件夹拖到目的硬盘驱动器的目录中。

通过 FTP 和互联网下载

下载这些文件的 FTP 地址是 **iplserv.ece.ucsb.edu**。针对 PC,Macintosh PowerPC 及 UNIX 工作站的文件目录如下:

pub/mitra/Labs/pc

pub/mitra/Labs/mac

pub/mitra/Labs/unix(只含有M文件)

按照以下步骤,可从FTP网址将文件下载到PC或UNIX电脑中:

1. 在电脑中运行FTP程序。
2. 输入 **ftp iplserv.ece.ucsb.edu**。
3. 在 **Login**:提示符处,输入 **anonymous**。
4. 在 **Password** 提示符处,输入你的电子邮件地址。
5. 输入 **cd pub/mitra/Labs/pc** 找到适用于PC的文件。输入 **cd pub/mitra/Labs/unix** 找到适用于UNIX的文件。这些目录中包含有我们要找的M文件和报告文件。
6. 输入 **cd** *目录名称*,进入存储M文件和报告文件的目录。
7. 输入 **asc**,下载ASCII文件(例如M文件)。输入 **bin**,下载二进制文件,例如Word报告文件(对PC)。输入 **get** *文件名* 下载所需要的文件类型。最后那条命令会将需要下载的文件存放在本地系统的当前目录中。
8. 使用 **get** 命令,下载其他的文件,或者输入 **cd**,回到前一级目录中。
9. 下载完所有的文件后,输入 **bye**。

将文件下载到Macintosh电脑中的FTP程序有多种。下面给出来自Dartmouth College的 **Fetch** 程序的下载步骤。

1. 运行该FTP程序。
2. 在 **New Connection** 对话框中输入下列信息。若看不到该对话框,可从 **File** 菜单中选择 **New Connection** 来打开它。

   `Host:` **iplserv.ece.ucsb.edu**

   `User ID:` **anonymous**

   `Password:` *你的电子邮件地址*

   `Directory` **/pub/mitra/Labs/mac**

3. 选定 **OK**。这时会出现一个显示FTP站点的目录内容的窗口。双击其他期望的文件夹可进入相应的目录。
4. 双击所要的文件可下载文件。这时会跳出一个对话框。在电脑上选择存储该文件的位置并选定 **Save**。在 **Fetch** 程序的最新版本中,选择目录名并选定 **Get** 后,就可下载整个目录。

通过互联网下载文件的步骤如下:

1. 打开可利用的网络浏览器。
2. 在URL窗口中输入 **ftp://iplserv.ece.ucsb.edu**。
3. 双击期望的目录,以进入目录(PC和Macintosh形式的目录如上所示)。
4. 双击所需要的文件进行下载。这时会跳出一个询问将文件存储至何处的对话框。

1.3 基础知识回顾

R1.1 离散时间信号用一个称为样本的数字序列来表示。典型的离散时间信号或序列$\{x[n]\}$的样本值用$x[n]$表示,其中自变量n的取值范围是从$-\infty$到∞之间的整数。为了表示方便,序列$\{x[n]\}$通常不用大括号表示。

R1.2 离散时间信号可以是一个有限长序列,也可以是一个无限长序列。有限长(也称为有限时宽或有限范围)序列仅定义在有限的时间间隔中:

$$N_1 \leqslant n \leqslant N_2 \tag{1.1}$$

其中$-\infty < N_1$而$N_2 < \infty$,$N_2 \geqslant N_1$。有限长序列的长度或时宽为

$$N = N_2 - N_1 + 1 \tag{1.2}$$

R1.3 满足

$$\tilde{x}[n] = \tilde{x}[n + kN] \quad 对所有\ n \tag{1.3}$$

的序列称之为周期为N的周期序列,其中N取任意正整数,k取任意整数。

R1.4 序列$x[n]$的能量定义为

$$\mathcal{E} = \sum_{n=-\infty}^{\infty} |x[n]|^2 \tag{1.4}$$

有限间隔$-K \leqslant n \leqslant K$上的序列能量定义为

$$\mathcal{E}_K = \sum_{n=-K}^{K} |x[n]|^2 \tag{1.5}$$

R1.5 非周期序列$x[n]$的平均功率定义为

$$\mathcal{P}_{av} = \lim_{K\to\infty} \frac{1}{2K+1}\mathcal{E}_K = \lim_{K\to\infty} \frac{1}{2K+1} \sum_{n=-K}^{K} |x[n]|^2 \tag{1.6}$$

周期为N的周期序列$\tilde{x}[n]$的平均功率为

$$\mathcal{P}_{av} = \frac{1}{N} \sum_{n=0}^{N-1} |\tilde{x}[n]|^2 \tag{1.7}$$

R1.6 单位样本序列(通常称为离散时间冲激或单位冲激)用$\delta[n]$表示,其定义为

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \tag{1.8}$$

单位阶跃序列用$\mu[n]$表示,其定义为

$$\mu[n] = \begin{cases} 1, & n \geqslant 0 \\ 0, & n < 0 \end{cases} \tag{1.9}$$

R1.7 指数序列由

$$x[n] = A\alpha^n \tag{1.10}$$

给定。其中A和α可以是任意实数或任意复数,表示为

$$\alpha = e^{(\sigma_o + j\omega_o)}, \qquad A = |A|e^{j\phi}$$

式(1.10)可改写为

$$x[n] = |A|e^{\sigma_o n + j(\omega_o n + \phi)} = |A|e^{\sigma_o n}\cos(\omega_o n + \phi) + j|A|e^{\sigma_o n}\sin(\omega_o n + \phi) \tag{1.11}$$

R1.8 带有常数振幅的实正弦序列形如

$$x[n] = A\cos(\omega_o n + \phi) \tag{1.12}$$

其中 A, ω_o 和 ϕ 是实数。在式(1.11)和式(1.12)中,参数 A, ω_o 和 ϕ 分别称为正弦序列 $x[n]$ 的振幅、角频率和初始相位。$f_o = \omega_o/2\pi$ 称为频率。

R1.9 若 $\omega_o N$ 是 2π 的整数倍,当 $\sigma_o = 0$ 时,式(1.11)的复指数序列和式(1.12)的正弦序列是周期序列,即

$$\omega_o N = 2\pi r \tag{1.13}$$

其中 N 为任意正整数,而 r 为任意整数。满足这种情况的最小 N 称为序列的周期。

R1.10 长度 N 的两个序列 $x[n]$ 和 $h[n]$ 的乘积,产生长度也为 N 的序列 $y[n]$

$$y[n] = x[n] \cdot h[n] \tag{1.14}$$

长度为 N 的两个序列 $x[n]$ 和 $h[n]$ 相加,产生长度也为 N 的序列 $y[n]$

$$y[n] = x[n] + h[n] \tag{1.15}$$

用标量 A 与长度为 N 的序列 $x[n]$ 相乘,得到长度为 N 的序列 $y[n]$

$$y[n] = A \cdot x[n] \tag{1.16}$$

无限长序列 $x[n]$ 通过时间反转,可得到无限长序列 $y[n]$

$$y[n] = x[-n] \tag{1.17}$$

无限长序列 $x[n]$ 经过 M 延时,可得到无限长序列 $y[n]$

$$y[n] = x[n - M] \tag{1.18}$$

若 M 是一个负整数,式(1.18)运算得到序列 $x[n]$ 的超前。

长度为 N 的序列 $x[n]$,可被长度为 M 的另一个序列 $g[n]$ 增补,得到长度为 $N+M$ 的更长序列 $y[n]$

$$\{y[n]\} = \{\{x[n]\}, \{g[n]\}\} \tag{1.19}$$

1.4 用到的 MATLAB 命令

这个练习中将用到的 MATLAB 命令如下:

运算符和特殊符号

```
:    .    +    -    *    /    ;    %
```

基本矩阵和矩阵控制

```
i    ones    pi    rand    randn    zeros
```

基本函数

```
cos     exp     imag     real
```

数据分析

```
sum
```

二维图形

```
axis    grid    legend    plot    stairs
stem    title   xlabel    ylabel
```

通用图形函数

```
clf     subplot
```

信号处理工具箱

```
sawtooth    square
```

关于这些命令的辅助说明，请参见 *MATLAB Reference Guide*[Mat94]和 *Signal Processing Toolbox User's Guide*[Mat96]，也可在命令窗口中键入 help *命令名称*。在附录 B 中可找到此处使用的 MATLAB 函数的简要解释。

1.5　序列的产生

本节的目的是使读者熟悉 MATLAB 中产生信号和绘制信号的基本命令。MATLAB 已被开发成能对数据向量或矩阵进行运算的工具。因此，序列以向量的形式存储，并且所有的信号被限定为因果的和有限长的。采用何种步骤执行本书中的程序，要根据 MATLAB 所运行的平台来决定。

基于 Windows PC 的 MATLAB

程序可通过在命令窗口中键入不带 .m 的程序名并按回车来执行。也可从命令窗口中的 **File** 菜单上，选择 **Open**，以选定所要的 M 文件，这将在 Edior/Debugger 窗口中打开这个 M 文件，使用 **Tools** 菜单下的 **Run** 命令，可运行该 M 文件。

基于 Macintosh 的 MATLAB

程序可通过在命令窗口中键入不带 .m 的程序名并按回车来执行。也可在屏幕上使用 **Open M-File** 命令，将该文件复制到 Editor 窗口，然后在屏幕上选择 **Save and Execute** 命令。

例 1.1　单位样本和单位阶跃序列

此例产生的两个基本离散时间序列分别是式(1.8)和式(1.9)给出的单位样本序列和单位阶跃序列。使用如下 MATLAB 命令，可产生长度为 N 的单位样本序列 u[n]

```
u = [1  zeros(1,N-1)];
```

延时 M 个样本且长度为 N 的单位样本序列 ud[n],可用如下的 MATLAB 命令产生,其中 M<N:

```
ud = [zero(1,M)  1  zeros(1,N-M-1)];
```

同样,使用下面的 MATLAB 命令可产生长度为 N 的单位阶跃序列 s[n]

```
s = [ones(1,N)];
```

产生延时单位阶跃序列的方法,类似于产生延时单位样本序列的方法。

程序 P1.1 产生并绘制了一个单位样本序列。

```
%程序 P1.1
%一个单位样本序列的产生
clf;
%产生从 -10 到 20 的一个向量
n = -10:20;
%产生单位样本序列
u = [zeros(1,10) 1 zeros(1,20)];
%绘制单位样本序列
stem(n,u);
xlabel('时间序号 n');ylabel('振幅');
title('单位样本序列');
axis([-10 20 0 1.2]);
```

习题

Q1.1 运行程序 P1.1,以产生单位样本序列 u[n]并显示它。

Q1.2 命令 clf,axis,title,xlabel 和 ylabel 的作用是什么?

Q1.3 修改程序 P1.1,以产生带有延时 11 个样本的延迟单位样本序列 ud[n]。运行修改的程序并显示产生的序列。

Q1.4 修改程序 P1.1,以产生单位步长序列 s[n]。运行修改后的程序并显示产生的序列。

Q1.5 修改程序 P1.1,以产生带有超前 7 个样本的延时单位阶跃序列 sd[n]。运行修改后的程序并显示产生的序列。

例 1.2 指数信号

另一种基本的离散时间序列是指数序列。此序列可使用 MATLAB 运算符 .^和 exp 产生。下面给出的程序 P1.2,可用来生成一个复数值的指数序列。

```
%程序 P1.2
%生成一个复指数序列
clf;
c = -(1/12)+(pi/6)*i;
K = 2;
n = 0:40;
x = K*exp(c*n);
subplot(2,1,1);
stem(n,real(x));
xlabel('时间序号 n');ylabel('振幅');
title('实部');
subplot(2,1,2);
```

```
stem(n,imag(x));
xlabel('时间序号 n');ylabel('振幅');
title('虚部');
```

下面给出的程序 P1.3,可用于生成一个实数值的指数序列。

```
% 程序 P1.3
% 生成一个实指数序列
clf;
n = 0:35; a = 1.2; K = 0.2;
x = K*a.^n;
stem(n,x);
xlabel('时间序号 n');ylabel('振幅');
```

习题

Q1.6　运行程序 P1.2,以产生复数值的指数序列。

Q1.7　哪个参数控制该序列的增长或衰减率？哪个参数控制该序列的振幅？

Q1.8　若参数 c 更改为(1/12)+(pi/6)*i,将会发生什么情况？

Q1.9　运算符 real 和 imag 的作用是什么？

Q1.10　命令 subplot 的作用是什么？

Q1.11　运行程序 P1.3,以产生实数值的指数序列。

Q1.12　哪个参数控制该序列的增长或衰减率？哪个参数控制该序列的振幅？

Q1.13　算术运算符^和 .^之间的区别是什么？

Q1.14　若参数 a 小于 1,会发生什么情况？将参数 a 更改为 0.9,将参数 k 更改为 20,再次运行程序 P1.3。

Q1.15　该序列的长度是多少？怎样才能改变它？

Q1.16　使用 MATLAB 命令 sum(s.*s)可计算用向量 s 表示的实数序列 s[n]的能量。试求在习题 Q1.11 和习题 Q1.14 中产生的实数值指数序列 x[n]的能量。

例 1.3　正弦序列

本例产生另一种非常有用的实正弦序列,该序列的形式如式(1.12)所示。这样的正弦序列在 MATLAB 中可使用三角运算符 cos 和 sin 产生。

程序 P1.4 是产生一个正弦信号的简单示例。

```
% 程序 P1.4
% 产生一个正弦序列
n = 0:40;
f = 0.1;
phase = 0;
A = 1.5;
arg = 2*pi*f*n - phase;
x = A*cos(arg);
clf; % 清除旧图形
stem(n,x); % 绘制产生的序列
axis([0  40  -2  2]);
grid;
```

```
title('正弦序列');
xlabel('时间序号 n');
ylabel('振幅');
axis;
```

习题

Q1.17 运行程序 P1.4,以产生正弦序列并显示它。

Q1.18 该序列的频率是多少? 怎样可以改变它? 哪个参数控制该序列的相位? 哪个参数控制该序列的振幅? 该序列的周期是多少?

Q1.19 该序列的长度是多少? 怎样可以改变它?

Q1.20 计算所产生的正弦序列的平均幂。

Q1.21 axis 和 grid 命令的作用是什么?

Q1.22 修改程序 P1.4,以产生一个频率为 0.9 的正弦序列并显示它。把此序列和习题 Q1.17 中产生的序列相比较。修改程序 P1.4,以产生一个频率为 1.1 的正弦序列并显示它。把此序列与习题 Q1.17 中产生的序列相比较,评价你的结果。

Q1.23 修改上述程序,以产生长度为 50、频率为 0.08、振幅为 2.5、相移为 90 度的一个正弦序列并显示它。该序列的周期是多少?

Q1.24 在程序 P1.4 中用 plot 命令代替 stem 命令,运行新程序。新图形与习题 Q1.17 中产生的图形有什么区别?

Q1.25 在程序 P1.4 中用 stairs 命令代替 stem 命令,运行新程序。新图形与习题 Q1.17 和习题 Q1.24 中产生的图形有什么区别?

例 1.4 随机信号

在区间(0,1)中均匀分布的长度为 N 的随机信号,可通过如下的 MATLAB 命令产生:

```
x = rand(1,N);
```

同理,使用下面的 MATLAB 命令,可产生长度为 N 且具有零均值和单位方差的正态分布的随机信号 x[n]

```
x = randn(1,N);
```

习题

Q1.26 编写一个 MATLAB 程序,以产生并显示一个长度为 100 的随机信号,该信号在区间 [-2,2]中均匀分布。

Q1.27 编写一个 MATLAB 程序,以产生并显示一个长度为 75 的高斯随机信号,该信号正态分布且均值为 0,方差为 3。

Q1.28 编写一个 MATLAB 程序,以产生并显示五个长度为 31 的随机正弦信号。

$$\{X[n]\} = \{A \cdot \cos(\omega_o n + \phi)\} \tag{1.20}$$

其中振幅 A 和相位ϕ是统计独立的随机变量,振幅在区间 $0 \leqslant A \leqslant 4$ 内均匀分布,相位在区间$0 \leqslant \phi \leqslant 2\pi$ 内均匀分布。

1.6 序列的简单运算

正如前面提到的,数字信号处理的目的,是从一个或多个给定的离散时间信号中,产生一个具有我们所需性质的信号。处理算法由诸如加法、标量乘法、时间反转、延时和乘积运算等基本运算的组合所组成(见R1.10)。在此,我们通过三个非常简单的例子来说明这些运算的应用。

例1.5 信号平滑

数字信号处理应用的一个常见例子是从被加性噪音污染的信号中移除噪声。假定信号 $s[n]$ 被噪声 $d[n]$ 所污染,得到了一个含有噪声的信号 $x[n]=s[n]+d[n]$。我们的目的是对 $x[n]$ 进行运算,产生一个合理逼近 $s[n]$ 的信号 $y[n]$。因此,对时刻 n 的样本附近的一些样本求平均,产生输出信号是一种简单有效的方法。例如,采用三点滑动平均算法的表达式如下:

$$y[n]=\frac{1}{3}(x[n-1]+x[n]+x[n+1]) \tag{1.21}$$

程序 P1.5 可用于实现上面的算法。

```
%程序 P1.5
%通过平均的信号平滑
clf;
R = 51;
d = 0.8*(rand(R,1) - 0.5); %产生随机噪声
m = 0:R-1;
s = 2*m.*(0.9.^m); %产生未污染的信号
x = s + d'; %产生被噪音污染的信号
subplot(2,1,1);
plot(m,d','r-',m,s,'g--',m,x,'b-.');
xlabel('时间序号 n');ylabel('振幅');
legend('d[n] ','s[n] ','x[n] ');
x1 = [0 0 x];x2 = [0 x 0];x3 = [x 0 0];
y = (x1 + x2 + x3)/3;
subplot(2,1,2);
plot(m,y(2:R+1),'r-',m,s,'g--');
legend('y[n] ','s[n] ');
xlabel('时间序号 n');ylabel('振幅');
```

习题

Q1.29 运行程序 P1.5,以产生所有相关的信号。

Q1.30 未污染的信号 s[n]是什么样的形式?加性噪声 d[n]是什么样的形式?

Q1.31 使用语句 x = s + d 能产生被噪声污染的信号吗?若不能,为什么?

Q1.32 信号 x1,x2 和 x3 与信号 x 之间的关系是什么?

Q1.33 legend 命令的作用是什么?

例1.6 复杂信号的产生

更复杂的信号可通过在简单信号上执行基本的运算来产生。例如,振幅调制信号可用低

频调制信号 $x_L=\cos(\omega_L n)$来调制高频正弦信号 $x_H[n]=\cos(\omega_H n)$。得到的信号 $y[n]$为

$$y[n]=A(1+m\cdot x_L[n])x_H[n]=A(1+m\cdot\cos(\omega_L n))\cos(\omega_H n)$$

其中 m 称为调制指数,用来确保$(1+m\cdot x_L[n])$在所有可能的 n 的情况下 m 都是正数。程序 P1.6 可用来产生一个振幅调制信号。

```
%程序 P1.6
%振幅调制信号的产生
n = 0:100;
m = 0.4;fH = 0.1; fL = 0.01;
xH = sin(2*pi*fH*n);
xL = sin(2*pi*fL*n);
y = (1+m*xL).*xH;
stem(n,y);grid;
xlabel('时间序号 n');ylabel('振幅');
```

习题

Q1.34 在载波信号 xH[n]和调制信号 xL[n]采用不同频率、不同调制指数 m 的情况下,运行程序 P1.6,以产生振幅调制信号 y[n]。

Q1.35 算术运算符 * 和 .* 之间的区别是什么?

由于正弦信号的频率是其相位对于时间的导数,为了产生频率随时间线性增加的扫频正弦信号,正弦信号的自变量必须是时间的二次函数。假定自变量的形式是 an^2+bn(即角频率为 $2an+b$)。在给出的条件(最小角频率和最大角频率)下,求出 a 和 b 的值。程序 P1.7 是产生这种信号的范例程序。

```
%程序 P1.7
%产生一个扫频正弦序列
n = 0:100;
a = pi/2/100;
b = 0;
arg = a*n.*n + b*n;
x = cos(arg);
clf;
stem(n,x);
axis([0,100,-1.5,1.5]);
title('扫频正弦信号');
xlabel('时间序号 n');
ylabel('振幅');
grid; axis;
```

习题

Q1.36 运行程序 P1.7,以产生扫频正弦序列 x[n]。

Q1.37 该信号的最小频率和最大频率是多少?

Q1.38 如何修改上述程序才能产生一个最小频率为 0.1、最大频率为 0.3 的扫频正弦信号?

1.7　工作区信息

命令 who 和 whos 可用于获得存储于工作区的所有变量的信息。

习题

Q1.39　在命令窗口中键入 who。此时在命令窗口中会显示什么信息？

Q1.40　在命令窗口中键入 whos。此时在命令窗口中会显示什么信息？

1.8　其他类型的信号(可选)

例 1.7　方波和锯齿波信号

MATLAB 函数 square 和 sawtooth 可分别用于产生图 1.1 和图 1.2 中显示的序列。

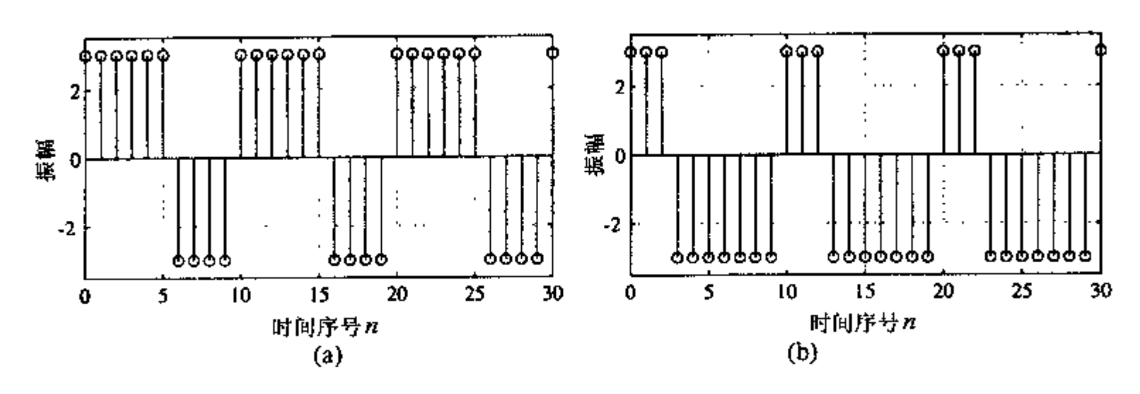


图 1.1　方波序列

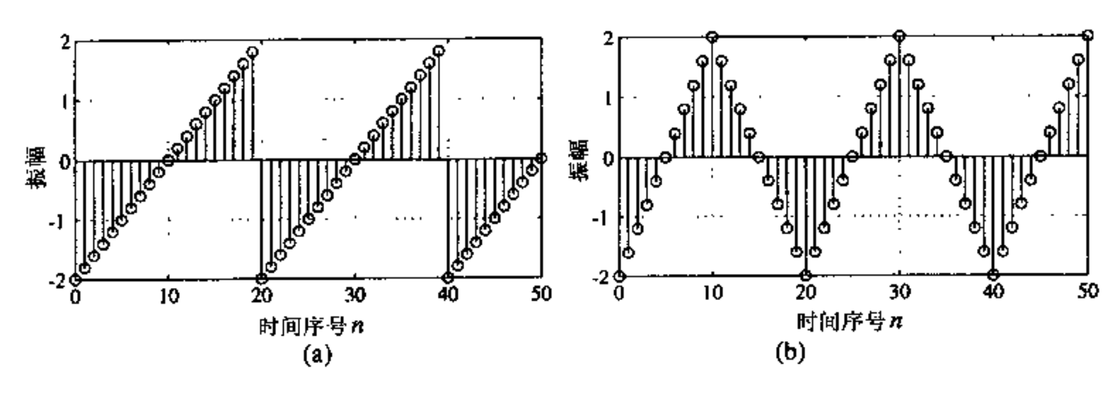


图 1.2　锯齿波序列

习题

Q1.41　编写 MATLAB 程序，以产生图 1.1 和图 1.2 中所示的方波和锯齿波序列，并将序列绘制出来。

1.9 基础知识阅读

[1] E. Cunningham. *Digital Filtering: An Introduction.* Houghton-Mifflin, Boston MA, 1992. Secs. 1.2–1.3.

[2] D. J. DeFatta, J. G. Lucas, and W. S. Hodgkiss. *Digital Signal Processing: A System Design Approach.* Wiley, New York NY, 1988. Secs. 2.1.2–2.1.4.

[3] L. B. Jackson. *Digital Filters and Signal Processing.* Kluwer, Boston MA, third edition, 1996. Secs. 2.2–2.3.

[4] R. Kuc. *Introduction to Digital Signal Processing.* McGraw-Hill, New York NY, 1988. Secs. 2-2, 2-4.

[5] L. C. Ludeman. *Fundamentals of Digital Signal Processing.* Harper & Row, New York NY, 1986. Sec. 1.2.

[6] S. K. Mitra. *Digital Signal Processing: A Computer-Based Approach.* McGraw-Hill, New York NY, 1998. Secs. 2.4–2.5.

[7] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing.* Prentice-Hall, Englewood Cliffs NJ, 1989. Sec. 2.1.

[8] B. Porat. *A Course in Digital Signal Procesing.* Wiley, New York NY, 1996. Sec. 2.7–2.8.

[9] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications.* Prentice-Hall, Englewood Cliffs NJ, 1996. Secs. 2.2–2.4.

[10] R. A. Roberts and C. T. Mullis. *Digital Signal Processing*, Addison-Wesley, Reading MA, 1987. Sec. 2.2.

第 2 章　离散时间系统的时域分析

1.2　导言

在时域中,离散时间系统对输入信号或者延迟信号进行简单运算处理,生成具有所需特性的输出信号。本章练习的目的就是通过 MATLAB 仿真一些简单的离散时间系统,并研究它们的时域特性。

2.2　基础知识回顾

R2.1　对线性离散时间系统,若 $y_1[n]$和 $y_2[n]$分别是输入序列 $x_1[n]$和 $x_2[n]$的响应,则输入

$$x[n]=\alpha x_1[n]+\beta x_2[n] \tag{2.1}$$

的输出响应为

$$y[n]=\alpha y_1[n]+\beta y_2[n] \tag{2.2}$$

式(2.2)的叠加性质对任意常量 α 和 β 以及任意输入 $x_1[n]$和 $x_2[n]$都成立。若存在一组非零的 α 和 β 值,或者一组非零的输入序列 $x_1[n]$和 $x_2[n]$,式(2.2)不成立,则系统称之为非线性的。

R2.2　对于离散时不变系统,若 $y_1[n]$是 $x_1[n]$的响应,则输入

$$x[n]=x_1[n-n_o]$$

的输出响应为

$$y[n]=y_1[n-n_o]$$

式中 n_o 是任意整数。上面的输入输出关系,对任意输入序列及其相应的输出成立。若对至少一个输入序列及其相应的输出序列不成立,则系统称之为时变的。

R2.3　线性时不变(LTI)系统既满足线性特性又满足时不变特性。

R2.4　若 $y_1[n]$和 $y_2[n]$分别是因果离散时间系统输入信号 $u_1[n]$和 $u_2[n]$的响应,则当

$$u_1[n]=u_2[n] \qquad n<N$$

时,有

$$y_1[n]=y_2[n] \qquad n<N$$

R2.5　若对于任意有界输入序列 $x[n]$,其输出 $y[n]$也是一个有界序列,则该离散时间系统是有界输入有界输出(BIBO)稳定的,也就是说,若

$$|x[n]|<B_x \quad \text{对所有 } n \text{ 值}$$

则相应的输出 $y[n]$也有界,即

$$|y[n]|<B_y \quad \text{对所有 } n \text{ 值}$$

这里 B_x 和 B_y 都是有限常数。

R2.6 数字滤波器对单位样本序列$\{\delta[n]\}$的响应称为单位样本响应,简称为冲激响应,用$\{h[n]\}$表示。相应地,离散时间系统对单位阶跃序列$\{\mu[n]\}$的响应用$\{s[n]\}$表示,称为单位阶跃响应,简称为阶跃响应。

R2.7 输入信号 $x[n]$的冲激响应 $h[n]$可用来描述线性时不变离散系统的响应 $y[n]$,表示为

$$y[n]=\sum_{k=-\infty}^{\infty}h[k]x[n-k] \tag{2.3}$$

通过简单的变量变化,可表示成

$$y[n]=\sum_{k=-\infty}^{\infty}h[n-k]x[k] \tag{2.4}$$

式(2.3)和式(2.4)中的和,称为序列 $x[n]$和 $h[n]$的卷积和,表示为

$$y[n]=h[n]\circledast x[n] \tag{2.5}$$

其中,符号$\circledast$表示卷积和。

R2.8 如图 2.1 所示,两个线性时不变离散时间系统的冲激响应分别为 $h_1[n]$和 $h_2[n]$,它们级联后的线性时不变离散时间系统的冲激响应 $h[n]$为

$$h[n]=h_1[n]\circledast h_2[n] \tag{2.6}$$

若图 2.1 中级联连接的两个线性时不变系统为

$$h_1[n]\circledast h_2[n]=\delta[n] \tag{2.7}$$

则线性时不变系统 $h_2[n]$称为线性时不变系统 $h_1[n]$的逆,反之亦然。

$\rightarrow$ $h_1[n]$ $\rightarrow$ $h_2[n]$ $\rightarrow$ $\equiv$ $\rightarrow$ $h_2[n]$ $\rightarrow$ $h_1[n]$ $\rightarrow$ $\equiv$ $\rightarrow$ $h_1[n]\circledast h_2[n]$ $\rightarrow$

图 2.1　级联连接

R2.9 当且仅当线性时不变离散时间系统的冲激响应序列$\{h[n]\}$绝对可和时,该线性时不变离散时间系统是 BIBO 稳定的,即

$$\sum_{n=-\infty}^{\infty}|h[n]|<\infty \tag{2.8}$$

R2.10 当且仅当线性时不变离散时间系统的冲激响应序列$\{h[n]\}$满足条件

$$h[k]=0\qquad k<0 \tag{2.9}$$

该线性时不变离散时间系统才是因果的。

R2.11 本书中,我们主要研究的线性时不变离散时间系统用形如

$$\sum_{k=0}^{N}d_k\,y[n-k]=\sum_{k=0}^{M}p_k\,x[n-k] \tag{2.10}$$

的线性常系数差分方程来描述。其中,$x[n]$和 $y[n]$分别为系统的输入和输出,$\{d_k\}$和$\{p_k\}$是常数。离散时间系统的阶数为 $\max(N,M)$,它表征系统差分方程的阶数。若假定系统是因果的,则可将式(2.10)改写,以将 $y[n]$表示为 $x[n]$的函数:

$$y[n] = -\sum_{k=1}^{N} \frac{d_k}{d_0} y[n-k] + \sum_{k=0}^{M} \frac{p_k}{d_0} x[n-k] \tag{2.11}$$

假设 $d_0 \neq 0$。已知 $x[n]$和初始条件 $y[n_o-1], y[n_o-2], \cdots, y[n_o-4]$,对所有 $n \geq n_o$ 的情况,可用式(2.11)计算输出 $y[n]$。

R2.12 若离散时间系统的冲激响应 $h[n]$是有限长的,则该系统称为有限冲激响应(FIR)系统。否则,该系统就称为无限冲激响应(IIR)系统。对于 $k>0$,若 $d_k=0$,式(2.11)表示的是一个有限冲激响应系统。否则,就为一个无限冲激响应系统。

2.3 用到的 MATLAB 命令

这个练习中要用到的 MATLAB 命令如下所示:

通用命令

disp

运算符和特殊字符

: . + - * / ; % <

语言构造与调试

break end for if input

基本矩阵和矩阵控制

ones pi zeros

基本函数

abs cos

多项式和内插函数

conv

二维图形

axis plot stem title xlabel ylabel

通用图形函数

clf subplot

字符串函数

num2str

信号处理工具箱

filter impz

关于这些命令的辅助说明,请参见 *MATLAB Reference Guide*[Mat94]和 *Signal Processing Toolbox User's Guide*[Mat96],或是在 Command 窗口中键入 help *命名名称*。在附录 B 中可找到此处使用的 MATLAB 函数的简要解释。

2.4 离散时间系统的仿真

在例 1.5 中,我们示例了一个应用,该应用使用式(1.21)给出的简单离散时间系统来平滑被随机噪声污染的数据。现在,我们通过仿真其他的一些离散时间系统来研究它们的特性。式(2.10)描述的因果线性时不变离散时间系统,可用命令 filter 进行仿真。该命令具有多种形式。若

```
num = [p0   p1   …   pM],
den = [d0   d1   …   dN],
```

则 y = filter(num,den,x)产生的输出向量 y 的长度与输入向量 x 的长度相同,并且初始值为零,即 y[-1] = y[-2] = … = y[-N] = 0。也可以用 y = filter(num,den,x,ic)计算系统输出,其中,ic = [y[-1],y[-2],…,y[-N]]是初始值。用[y,fc] = filter(num,den,x,ic)可得到最终值。

例 2.1 滑动平均系统

式(1.21)表明,三点平滑滤波器是一个线性时不变的有限冲激响应系统。并且,由于 $y[n]$依赖于输入样本 $x[n+1]$的未来值,因此该系统是非因果的。将输出延时一个抽样周期,可得到三点平滑滤波器的因果表达式,生成的 FIR 滤波器表示为

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2]) \tag{2.12}$$

归纳上式可得

$$y[n] = \frac{1}{M}\sum_{k=0}^{M-1} x[n-k] \tag{2.13}$$

此式表示了一个因果 M 点平滑 FIR 滤波器。式(2.13)表示的系统即是一个滑动平均滤波器。下面我们通过从若干个正弦信号之和所组成的信号中滤出高频分量,来讲解该滑动平均滤波器的使用。

```
%程序 P2.1
%一个 M 点滑动平均滤波器的仿真
%产生输入信号
clf;
n = 0:100;
s1 = cos(2*pi*0.05*n); %一个低频正弦
s2 = cos(2*pi*0.47*n); %一个高频正弦
x = s1+s2;
%滑动平均滤波器的实现
M = input('滤波器所需的长度 = ');
num = ones(1,M);
y = filter(num,1,x)/M;
```

```
%计算输出信号
x1 = [x 0 0]; %x1[n] = x[n+1]
x2 = [0 x 0]; %x2[n] = x[n]
x3 = [0 0 x]; %x3[n] = x[n-1]
y = x2.*x2-x1.*x3;
y = y(2:202);
%画出输入和输出信号
subplot(2,1,1)
plot(n,x)
xlabel('时间序号 n');ylabel('振幅');
title('输入信号')
subplot(2,1,2)
plot(n,y)
xlabel('时间序号 n');ylabel('振幅');
title('输出信号');
```

习题

Q2.5 用不同频率的正弦信号作为输入信号,计算每个输入信号的输出信号。输出信号是如何受到输入信号频率的影响的?你能在数学上对你的结论加以证明吗?

Q2.6 用形如 $x[n]=\sin(\omega_o n)+K$ 的正弦信号作为输入信号,求出系统的输出信号。输出信号 $y[n]$是如何受到 DC 值 K 的影响的?

例 2.3 线性和非线性系统

现在我们来研究式(2.10)描述的因果系统的线性特性(见 R2.1)。假定系统为

$$y[n]-0.4y[n-1]+0.75y[n-2]=2.2403x[n]+2.4908x[n-1]+2.2403x[n-2] \qquad (2.15)$$

用 MATLAB 程序 P2.3 仿真式(2.15)描述的系统,输入三个不同的输入序列 $x_1[n]$,$x_2[n]$和 $x[n]=a\cdot x_1[n]+b\cdot x_2[n]$,计算并求出相应的输出响应 $y_1[n]$,$y_2[n]$和 $y[n]$。

```
%程序 P2.3
%生成输入序列
clf;
n = 0:40;
a = 2;b = -3;
x1 = cos(2*pi*0.1*n);
x2 = cos(2*pi*0.4*n);
x = a*x1 + b*x2;
num = [2.2403 2.4908 2.2403];
den = [1 -0.4 0.75];
ic = [0 0]; %设置零初始条件
y1 = filter(num,den,x1,ic); %计算输出 y1[n]
y2 = filter(num,den,x2,ic); %计算输出 y2[n]
y = filter(num,den,x,ic); %计算输出 y[n]
yt = a*y1 + b*y2;
d = y - yt; %计算差值输出 d[n]
%画出输出和差信号
subplot(3,1,1)
stem(n,y);
ylabel('振幅');
```

```
title('加权输入: a \cdot x_{1}[n] + b \cdot x_{2}[n]的输出');
subplot(3,1,2)
stem(n,yt);
ylabel('振幅');
title('加权输出 t: a \cdot y_{1}[n] + b \cdot y_{2}[n]');
subplot(3,1,3)
stem(n,d);
xlabel('时间序号 n');ylabel('振幅');
title('差信号');
```

习题

Q2.7 运行程序 P2.3,对由加权输入得到的 y[n]与在相同权系数下输出 y1[n]和 y2[n]相加得到的 yt[n]进行比较,这两个序列是否相等? 该系统是线性系统吗?

Q2.8 用三组不同的权系数 a 和 b 的值以及三组不同的输入频率重做习题 Q2.7。

Q2.9 当初始条件非零时重做习题 Q2.7。

Q2.10 当初始条件非零时重做习题 Q2.8。

Q2.11 假定另一个系统为

$$y[n] = x[n]x[n-1]$$

修改程序 P2.3,计算这个系统的输出序列 y1[n],y2[n]和 y[n]。比较 y[n]和 yt[n]。这两个序列是否相等? 该系统是线性系统吗?

例 2.4　时不变系统和时变系统

下面我们来研究式(2.11)描述的因果系统的时不变特性(见 R2.2)。重新考虑式(2.15)描述的系统。

用 MATLAB 程序 P2.4 仿真式(2.15)给出的系统,以产生两个不同的输入序列 x[n]和 x[n-D],计算并画出相应的输出序列 y1[n],y2[n]和 y1[n]-y2[n+D]。

```
% 程序 P2.4
% 生成输入序列
clf;
n = 0:40; D = 10;a = 3.0;b = -2;
x = a*cos(2*pi*0.1*n) + b*cos(2*pi*0.4*n);
xd = [zeros(1,D) x];
num = [2.2403 2.4908 2.2403];
den = [1 -0.4 0.75];
ic = [0 0]; % 设置初始条件
% 计算输出 y[n]
y = filter(num,den,x,ic);
% 计算输出 yd[n]
yd = filter(num,den,xd,ic);
% 计算差值输出 d[n]
d = y - yd(1+D:41+D);
% 画出输出
subplot(3,1,1)
stem(n,y);
ylabel('振幅');
```

```
title('输出 y[n]'); grid;
subplot(3,1,2)
stem(n,yd(1:41));
ylabel('振幅');
title(['由于延时输入 x[n ',num2str(D),']的输出']); grid;
subplot(3,1,3)
stem(n,d);
xlabel('时间序号 n'); ylabel('振幅');
title('差值信号'); grid;
```

习题

Q2.12 运行程序 P2.4 并比较输出序列 y[n]和 yd[n-10]。这两个序列之间有什么关系?该系统是时不变系统吗?

Q2.13 采用三个不同的延时变量 D 的值重做习题 Q2.12。

Q2.14 采用三组不同的输入频率的值重做习题 Q2.12。

Q2.15 在非零的初始条件下重做习题 Q2.12。该系统是时不变系统吗?

Q2.16 在非零的初始条件下重做习题 Q2.14。该系统是时不变系统吗?

Q2.17 考虑另一个系统:

$$y[n] = n\,x[n] + x[n-1] \tag{2.16}$$

修改程序 P2.4,以仿真上面的系统并确定该系统是否为时不变系统。

Q2.18 (可选)修改程序 P2.3 来测试式(2.16)给出的系统的线性性。

2.5 线性时不变离散时间系统

例 2.5 线性时不变系统的冲激响应的计算

MATLAB 命令 y = impz(num,den,N),可用来计算式(2.11)所示的因果线性时不变离散时间系统的冲激响应的前 N 个样本。

下面给出的 MATLAB 程序 P2.5 计算并绘出了式(2.15)描述的系统的冲激响应。

```
% 程序 P2.5
% 计算冲激响应 y
clf;
N = 40;
num = [2.2403 2.4908 2.2403];
den = [1 -0.4 0.75];
y = impz(num,den,N);
% 画出冲激响应
stem(y);
xlabel('时间序号 n'); ylabel('振幅');
title('冲激响应'); grid;
```

习题

Q2.19 运行程序 P2.5,生成式(2.15)所给离散时间系统的冲激响应。

Q2.20　修改程序 P2.5,产生如下因果线性时不变系统的冲激响应的前 45 个样本：

$$\begin{aligned} &y[n]+0.71\,y[n-1]-0.46\,y[n-2]-0.62\,y[n-3] \\ &\quad =0.9\,x[n]-0.45\,x[n-1]+0.35\,x[n-2]+0.002\,x[n-3] \end{aligned} \tag{2.17}$$

Q2.21　利用 filter 命令编写一个 MATLAB 程序,生成式(2.17)给出的因果线性时不变系统的冲激响应,计算并画出前 40 个样本。把你的结果和习题 Q2.20 中得到的结果相比较。

Q2.22　编写一个 MATLAB 程序,生成并画出式(2.11)给出的因果线性时不变系统的阶跃响应。用该程序计算并画出式(2.15)给出的线性时不变系统的阶跃响应的前 40 个样本。

例 2.6　线性时不变系统的级联

在实际应用中,高阶因果线性时不变离散时间系统可以用低阶因果线性时不变离散时间系统级联得到。例如,四阶离散时间系统

$$\begin{aligned} &y[n]+1.6\,y[n-1]+2.28\,y[n-2]+1.325\,y[n-3]+0.68\,y[n-4] \\ &\quad =0.06\,x[n]-0.19\,x[n-1]+0.27\,x[n-2]-0.26\,x[n-3]+0.12\,x[n-4] \end{aligned} \tag{2.18}$$

可用两个二阶离散时间系统的级联实现:

第一级

$$y_1[n]+0.9\,y_1[n-1]+0.8\,y_1[n-2]=0.2\,x[n]-0.2\,x[n-1]+0.4\,x[n-2] \tag{2.19}$$

第二级

$$y_2[n]+0.7\,y_2[n-1]+0.85\,y_2[n-2]=0.2\,y_1[n]-0.5\,y_1[n-1]+0.3\,y_1[n-2] \tag{2.20}$$

MATLAB 程序 P2.6 分别仿真了式(2.18)给出的四阶系统和式(2.19)及式(2.20)给出的级联系统。该程序首先产生了序列 x[n],把它作为四阶系统的输入,生成输出 y[n]。然后,将同样的输入 x[n]应用到第一级得到输出序列 y1[n]。接下来,用 y1[n]作为第二级的输入得到它的输出 y2[n]。最后,求出 y[n]和 y2[n]之间的差,并画出所有的输出信号和差值信号。

```
%程序 P2.6
%级联实现
clf;
x = [1 zeros(1,40)]; %生成输入
n = 0:40;
%四阶系统的系数
den = [1  1.6  2.28  1.325  0.68];
num = [0.06  -0.19  0.27  -0.26  0.12];
%计算四阶系统的输出
y = filter(num,den,x);
%两个二阶系统的系数
num1 = [0.3   -0.2   0.4];den1 = [1   0.9   0.8];
num2 = [0.2   -0.5   0.3];den2 = [1   0.7   0.85];
%级联第一级的输出 y1[n]
y1 = filter(num1,den1,x);
%级联第二级的输出 y2[n]
y2 = filter(num2,den2,y1);
```

```
%y[n]和y2[n]之间的差
d = y - y2;
%画出输出和差值信号
subplot(3,1,1);
stem(n,y);
ylabel('振幅');
title('四阶实现的输出'); grid;
subplot(3,1,2);
stem(n,y2)
ylabel('振幅');
title('级联实现的输出'); grid;
subplot(3,1,3);
stem(n,d)
xlabel('时间序号 n');ylabel('振幅');
title('差值信号'); grid;
```

习题

Q2.23 运行程序 P2.6,计算输出序列 y[n]和 y2[n]以及差值信号 d[n]。y[n]和 y2[n]相等吗?

Q2.24 将输入改成一个正弦序列,重做习题 Q2.23。

Q2.25 用任意的非零初始向量 ic,ic1 和 ic2 来重做习题 Q2.23。

Q2.26 修改程序 P2.6,将两个二阶系统顺序颠倒并在零初始条件下重复此过程。两个输出之间有差别吗?

Q2.27 用任意非零初始向量 ic,ic1 和 ic2 来重做习题 Q2.26。

例 2.7 卷积

假设待卷积的两个序列都为有限长序列,式(2.5)中的卷积运算符在 MATLAB 中可通过命令 conv 实现。例如,可以把系统的冲激响应与给定的有限长输入序列进行卷积,得到有限冲激响应系统的输出序列。下面的 MATLAB 程序实现了该方法。

```
%程序 P2.7
clf;
h = [3  2  1  -2  1  0  -4  0  3]; %冲激
x = [1  -2  3  -4  3  2  1]; %输入序列
y = conv(h,x);
n = 0:14;
subplot(2,1,1);
stem(n,y);
xlabel('时间序号 n'); ylabel('振幅');
title('用卷积得到的输出'); grid;
x1 = [x zeros(1,8)];
y1 = filter(h,1,x1);
subplot(2,1,2);
stem(n,y1);
xlabel('时间序号 n'); ylabel('振幅');title('由滤波生成的输出'); grid;
```

习题

Q2.28 运行程序 P2.7,对序列 h[n]和 x[n]求卷积,生成 y[n],并用 FIR 滤波器 h[n]对输入

x[n]滤波,求得 y1[n]。y[n]和 y1[n]有差别吗?为什么要使用对 x[n]补零后得到的 x1[n]作为输入来产生 y1[n]?

Q2.29 修改程序 P2.7,计算长度为 15 的序列 h[n]和长度为 10 的序列 x[n]的卷积,重做问题 Q2.28。h[n]和 x[n]的样本值由你自己给定。

例 2.8　线性时不变系统的稳定性

如式(2.8)所示,若一个线性时不变离散时间系统的冲激响应是绝对可和的,则该系统就是 BIBO 稳定的。因此可知,无限冲激响应线性时不变系统稳定的一个必要条件是,随着样本的增加,冲激响应衰减到零。程序 P2.8 给出的 MATLAB 程序,计算了一个因果 IIR 线性时不变系统的冲激响应的绝对值的和。它计算了冲激响应序列的 N 个样本,计算持续增加的 K 值的表达式为

$$S(K)=\sum_{n=0}^{K}|h[n]| \tag{2.21}$$

在每一次迭代中检查 $|h(K)|$ 的值。若 $|h[K]|$ 的值小于 10^{-6},可认为式(2.21)中的 $S(K)$ 已经收敛并且非常接近于 $S(\infty)$。

```
% 程序 P2.8
% 基于冲激响应样本绝对值
% 的和的稳定性测试
clf;
num = [1  -0.8]; den = [1  1.5  0.9];
N = 200;
h = impz(num,den,N+1);
parsum = 0;
for k = 1:N+1;
    parsum = parsum + abs(h(k));
    if abs(h(k)) < 10^(-6),break,end
end
% 画出冲激响应
n = 0:N;
stem(n,h)
xlabel('时间序号 n'); ylabel('振幅');
disp('值 =');disp(abs(h(k))); % 显示 abs(h(k))的值
```

习题

Q2.30 程序中为什么要使用命令 for 和 end?

Q2.31 使用命令 break 的目的是什么?

Q2.32 程序 P2.8 确定的冲激响应所对应的离散时间系统是什么?运行程序 P2.8,生成该冲激响应。该系统稳定吗?若 $|h[K]|$ 不小于 10^{-6} 且图形显示了一个衰减的冲激响应,用一个较大的 N 值再次运行程序 P2.8。

Q2.33 考虑用差分方程

$$y[n]=x[n]-4x[n-1]+3x[n-2]+1.7y[n-1]-y[n-2]$$

描述的离散时间系统。修改程序 P2.8,计算并画出上述系统的冲激响应。该系统稳定吗?

例 2.9 滤波概念的解释

考虑用如下差分方程描述的两个离散时间系统:

系统 1

$$y[n]=0.5x[n]+0.27x[n-1]+0.77x[n-2]$$

系统 2

$$y[n]=0.45x[n]+0.5x[n-1]+0.45x[n-2]+0.53y[n-1]-0.46y[n-2]$$

对输入

$$x[n]=\cos\left(\frac{20\pi n}{256}\right)+\cos\left(\frac{200\pi n}{256}\right),\qquad 0\leqslant n<299$$

用 MATLAB 程序 P2.9 来计算上述两个系统的输出。

```
% Program P2.9
% 产生输入序列
clf;
n = 0:299;
x1 = cos(2*pi*10*n/256);
x2 = cos(2*pi*100*n/256);
x = x1 + x2;
% 计算输出序列
num1 = [0.5 0.27 0.77];
y1 = filter(num1,1,x); % 系统 #1 的输出
den2 = [1 -0.53 0.46];
num2 = [0.45 0.5 0.45];
y2 = filter(num2,den2,x); % 系统 #2 的输出
% 画出输出序列
subplot(2,1,1);
plot(n,y1);axis([0 300 -2 2]);
ylabel('振幅');
title('系统 #1 的输出'); grid;
subplot(2,1,2);
plot(n,y2);axis([0 300 -2 2]);
xlabel('时间序号 n'); ylabel('振幅');
title('系统 #2 的输出'); grid;
```

习题

Q2.34 运行程序 P2.9。在该程序中,两个滤波器都是低通滤波器,在阻带中,尤其是在输入信号的频点上,有不同的衰减。哪个滤波器能更好地抑制输入信号 x[n]的高频分量?

Q2.35 修改程序 P2.9,将输入序列改变成扫频正弦序列(长度为 301、最低频率为 0、最高频率为 0.5)。哪个滤波器能更好地抑制输入信号 x[n]的高频分量?

2.6 基础知识阅读

[1] A. Antoniou. *Digital Filters: Analysis, Design, and Applications*. McGraw-Hill, New York NY, second edition, 1993. Secs. 1.3 – 1.8.

[2] E. Cunningham. *Digital Filtering: An Introduction*. Houghton-Mifflin, New York NY, 1992. Ch. 8.

[3] D.J. DeFatta, J.G. Lucas, and W.S. Hodgkiss. *Digital Signal Processing: A System Design Approach*. Wiley, New York NY, 1988. Sec. 2.2.

[4] L.B. Jackson. *Digital Filters and Signal Processing*. Kluwer, Boston MA, third edition, 1996. Ch. 11.

[5] R. Kuc. *Introduction to Digital Signal Processing*. McGraw-Hill, New York NY, 1988. Secs. 2-3, 2-5 – 2-8.

[6] L.C. Ludeman. *Fundamentals of Digital Signal Processing*, Harper & Row, New York NY, 1986. Sec. 1.3.

[7] S.K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, New York NY, 1998. Sec. 8.4 and Ch. 9.

[8] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs NJ, 1989. Secs. 2.2–2.5.

[9] S.J. Orfanidis. *Introduction to Signal Processing*. Prentice-Hall, Englewood Cliffs NJ, 1996. Chs. 2, 3.

[10] B. Porat. *A Course in Digital Signal Procesing*. Wiley, New York, NY, 1996. Sec. 2.7.

[11] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs NJ, third edition, 1996. Secs. 2.2–2.4.

[12] R.A. Roberts and C.T. Mullis. *Digital Signal Processing*. Addison-Wesley, Reading MA, 1987. Secs. 2.3–2.4.

第3章　离散时间信号的频域分析

3.1　导言

在前面的两个练习中，我们学习了离散时间信号与系统的时域分析，并对它们的性质做了研究。对这些信号和系统在频域中进行分析，可以进一步研究它们的性质。频域中最常见的三种表示是离散时间傅里叶变换（DTFT）、离散傅里叶变换（DFT）和 z 变换。在本章的练习中，我们将学习离散时间序列的所有三种表示方法。

3.2　基础知识回顾

R3.1　序列 $x[n]$ 的离散时间傅里叶变换（DTFT）$X(e^{j\omega})$ 定义为

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \tag{3.1}$$

通常 $X(e^{j\omega})$ 是实变量 ω 的一个复函数并可写为

$$X(e^{j\omega}) = X_{re}(e^{j\omega}) + jX_{im}(e^{j\omega}) \tag{3.2}$$

其中 $X_{re}(e^{j\omega})$ 和 $X_{im}(e^{j\omega})$ 分别是 $X(e^{j\omega})$ 的实部和虚部，它们都是 ω 的实函数。$X(e^{j\omega})$ 也可表示为

$$X(e^{j\omega}) = |X(e^{j\omega})|e^{j\theta(\omega)} \tag{3.3}$$

其中，

$$\theta(\omega) = \arg\{X(e^{j\omega})\} \tag{3.4}$$

$|X(e^{j\omega})|$ 称为幅度函数，$\theta(\omega)$ 称为相位函数，这两个函数都是 ω 的实函数。在很多应用中，傅里叶变换称为傅里叶谱，而 $|X(e^{j\omega})|$ 和 $\theta(\omega)$ 分别称为幅度谱和相位谱。

R3.2　离散时间傅里叶变换 $X(e^{j\omega})$ 是一个周期为 2π 的 ω 的连续函数。

R3.3　对实序列 $x[n]$，其离散时间傅里叶变换的实部 $X_{re}(e^{j\omega})$ 及其幅度函数 $|X(e^{j\omega})|$ 都是 ω 的偶函数，虚部 $X_{im}(e^{j\omega})$ 和相位函数 $\theta(\omega)$ 都是 ω 的奇函数。

R3.4　$X(e^{j\omega})$ 的离散时间傅里叶逆变换 $x[n]$ 为

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega \tag{3.5}$$

R3.5　若 $x[n]$ 绝对可和，即

$$\sum_{n=-\infty}^{\infty} |x[n]| < \infty \tag{3.6}$$

则序列 $x[n]$ 的傅里叶变换 $X(e^{j\omega})$ 存在。

$$h_e[n]=\begin{cases} h[n], & 0\leqslant n\leqslant M-1 \\ 0, & M\leqslant n\leqslant N+M-1 \end{cases} \tag{3.14}$$

R3.11 离散傅里叶变换满足很多有用的性质,这些性质在很多应用中都会用到。这些性质以及它们的分析证明过程,我们可在任何一本数字信号处理的课本中找到。这些性质也可以用 MATLAB 来证明。下面我们列出了在本练习中将会遇到的几个性质。

圆周时移性质:若 $G[k]$表示长度为 N 的序列 $g[n]$的 N 点离散傅里叶变换,则圆周时移序列 $g[\langle n-n_o\rangle_N]$的 N 点离散傅里叶变换为 $W_N^{kn_o}G[k]$,其中 $W_N=e^{-j2\pi/N}$。

圆周频移性质:若 $G[k]$表示长度为 N 的序列 $g[n]$的 N 点离散傅里叶变换,则序列 $W_N^{-k_on}g[n]$的 N 点离散傅里叶变换为 $G[\langle k-k_o\rangle_N]$。

圆周卷积性质:若 $G[k]$和 $H[k]$分别表示长度为 N 的序列 $g[n]$和 $h[n]$的 N 点离散傅里叶变换,则圆周卷积序列 $g[n]$Ⓝ$h[n]$的 N 点离散傅里叶变换为 $G[k]H[k]$。

帕斯瓦尔关系:若 $G[k]$表示长度为 N 的序列 $g[n]$的 N 点离散傅里叶变换,则

$$\sum_{n=0}^{N-1}|g[n]|^2=\frac{1}{N}\sum_{k=0}^{N-1}|G[k]|^2 \tag{3.15}$$

R3.12 长度为 N 的实序列 $g[n]$,其周期偶函数部分 $g_{pe}[n]$和周期奇函数部分 $g_{po}[n]$为

$$g_{pe}[n]=\frac{1}{2}(g[n]+g[\langle -n\rangle_N]) \tag{3.16}$$

$$g_{po}[n]=\frac{1}{2}(g[n]-g[\langle -n\rangle_N]) \tag{3.17}$$

若 $G[k]$表示长度为 N 的序列 $g[n]$的 N 点 DFT 变换,则 $g_{pe}[n]$和 $g_{po}[n]$的 N 点离散傅里叶变换为 $Re\{G[k]\}$和 $j\,Im\{G[k]\}$。

R3.13 若 $g[n]$和 $h[n]$表示两个长度为 N 的实序列,$G[k]$和 $H[k]$表示它们各自的 N 点离散傅里叶变换。定义 $x[n]=g[n]+jh[n]$,那么 $x[n]$为长度为 N 的复数序列,对 $x[n]$求它的 N 点离散傅里叶变换 $X[k]$,我们可以通过下式有效地计算 $G[k]$和 $H[k]$:

$$G[k]=\frac{1}{2}(X[k]+X^*[\langle -k\rangle_N]) \tag{3.18}$$

$$H[k]=\frac{1}{2j}(X[k]-X^*[\langle -k\rangle_N]) \tag{3.19}$$

R3.14 设 $v[n]$是长度为 $2N$ 的实序列,$V[k]$表示它的 $2N$ 点离散傅里叶变换。将长度为 N 的两个实序列 $g[n]$和 $h[n]$分别定义为

$$g[n]=v[2n]\ \ and\ \ h[n]=v[2n+1],\quad 0\leqslant n<N \tag{3.20}$$

并且 $G[k]$和 $H[k]$表示它们的 N 点离散傅里叶变换,则 $v[n]$的 $2N$ 点离散傅里叶变换 $V[k]$,可以自两个 N 点离散傅里叶变换 $G[k]$和 $H[k]$计算得到,即

$$V[k]=G[\langle k\rangle_N]+W_{2N}^kH[\langle k\rangle_N],\quad 0\leqslant k\leqslant 2N-1 \tag{3.21}$$

R3.15 序列 $g[n]$的 z 变换 $G(z)$定义为

$$G(z)=\mathcal{Z}\{g[n]\}=\sum_{n=-\infty}^{\infty}g[n]z^{-n} \tag{3.22}$$

其中 z 是复变量。使得 z 变换 $G(z)$ 收敛的一组 z 的值 $\mathscr{R}$ 称为它的收敛域(ROC)。通常，序列 $g[n]$ 的 z 变换的收敛域 $\mathscr{R}$ 是在 z 平面内的一个环形区域：

$$R_{g-} < |z| < R_{g+} \tag{3.23}$$

其中 $0 \leqslant R_{g-} < R_{g+} \leqslant \infty$。

R3.16 在线性时不变离散时间系统中，所有有关的 z 变换都是 z^{-1} 的有理函数，即两个 z^{-1} 多项式的比值：

$$G(z) = \frac{P(z)}{D(z)} = \frac{p_0 + p_1 z^{-1} + \ldots + p_{M-1} z^{-(M-1)} + p_M z^{-M}}{d_0 + d_1 z^{-1} + \ldots + d_{N-1} z^{-(N-1)} + d_N z^{-N}} \tag{3.24}$$

它可重写为

$$G(z) = \frac{p_0}{d_0} \frac{\prod_{r=1}^{M}(1 - \xi_r z^{-1})}{\prod_{s=1}^{N}(1 - \lambda_s z^{-1})} = \frac{p_0}{d_0} z^{N-M} \frac{\prod_{r=1}^{M}(z - \xi_r)}{\prod_{s=1}^{N}(z - \lambda_s)} \tag{3.25}$$

其中 $G(z)$ 的零点为 $z = \xi_r$，$G(z)$ 的极点为 $z = \lambda_s$。当 $N > M$ 时，在 $z = 0$ 处有 $(N - M)$ 个零点(z 平面的原点)；当 $N < M$ 时，在 $z = 0$ 处有 $(N - M)$ 个极点。

R3.17 一个序列若具有有理 z 变换，则其 z 变换的收敛域不能包含任何极点，并且极点的位置限制了收敛域的范围。

R3.18 z 变换 $G(z)$ 的逆 z 变换 $g[n]$ 为

$$g[n] = \frac{1}{2\pi j} \oint_C G(z)\, z^{n-1}\, dz \tag{3.26}$$

其中 C 是 $G(z)$ 的收敛域内围绕 $z = 0$ 逆时针旋转的曲线。

R3.19 有理 z 变换 $G(z) = P(z)/D(z)$[其中多项式 $P(z)$ 的最高次数为 M，多项式 $D(z)$ 的最高次数为 N，并在 $z = \lambda_s$ 处有不同的极点，$s = 1, 2, \cdots, N$]的部分分式展开形式为

$$G(z) = \sum_{\ell=0}^{M-N} \eta_\ell z^{-\ell} + \sum_{s=0}^{N} \frac{\rho_s}{1 - \lambda_s z^{-1}} \tag{3.27}$$

假定 $M \geqslant N$。上面的表达式中的常数 ρ_s 称为留数，为

$$\rho_s = (1 - \lambda_s z^{-1}) G(z)|_{z=\lambda_s} \tag{3.28}$$

若 $G(z)$ 有多重极点，部分分式展开式有稍微不同的形式。例如，若在 $z = \nu$ 处有 L 重极点，并且余下的 $N - L$ 个极点分别是在 $z = \lambda_s$ 处的单极点，$s = 1, 2, \cdots, N - L$，此时 $G(z)$ 的部分分式表达形式为

$$G(z) = \sum_{\ell=0}^{M-N} \eta_\ell z^{-\ell} + \sum_{s=0}^{N-L} \frac{\rho_s}{1 - \lambda_s z^{-1}} + \sum_{r=1}^{L} \frac{\gamma_r}{(1 - \nu z^{-1})^r} \tag{3.29}$$

其中常数 γ_r(当 $r \neq 1$ 时不再称为留数)用公式

$$\gamma_r = \frac{1}{(L-r)!(-\nu)^{L-r}} \frac{d^{L-r}}{d(z^{-1})^{L-r}} \left[(1 - \nu z^{-1})^L G(z)\right]_{z=\nu}, \quad r = 1, \cdots, L \tag{3.30}$$

计算，留数 ρ_s 仍用式(3.28)计算。

3.3　用到的 MATLAB 命令

这次练习中要用到的 MATLAB 命令如下：

通用命令

```
disp
```

运算符和特殊字符

```
:    .    +    -    *    /    ;
%    <    >    .*   ^    .^   =
```

语言构造与调试

```
break    end    error    for    function    if    input    pause
```

基本矩阵和矩阵控制

```
fliplr    i    pi    zeros    :
```

基本函数

```
abs    angle    conj    exp    imag    real    rem
```

多项式和内插函数

```
conv
```

二维图形

```
axis    grid    plot    stem    title    xlabel    ylabel
```

通用图形函数

```
clf    subplot
```

字符串函数

```
num2str
```

数据分析和傅里叶变换函数

```
fft    ifft    max    min
```

信号处理工具箱

```
freqz    impz    residuez    tf2zp    zp2sos    zp2tf    zplane
```

关于这些命令的辅助说明，请参见 *MATLAB Reference Guide*[Mat94]和 *Signal Processing Toolbox User's Guide*[Mat96]，或在命令窗口键入 `help` *命令名称*。在附录 B 中可找到此处使用的 MATLAB 函数的简要解释。

3.4　离散时间傅里叶变换

序列 $x[n]$的离散时间傅里叶变换 $X(e^{j\omega})$是 ω 的连续函数。由于数据在 MATLAB 中以向量的形式存在，$X(e^{j\omega})$只能在一个给定的离散频率的集合中计算。然而，只有类似

$$X(e^{j\omega})=\frac{p_0+p_1e^{-j\omega}+\cdots+p_Me^{-j\omega M}}{d_0+d_1e^{-j\omega}+\cdots+d_Ne^{-j\omega N}} \tag{3.31}$$

形式的 $e^{-j\omega}$的有理函数，才能计算其离散时间傅里叶变换。在下面两个练习中，我们将学习怎样用 MATLAB 来求取并画出序列的离散时间傅里叶变换，并学习离散时间傅里叶变换的一些特性。

例 3.1　计算离散时间傅里叶变换

如式(3.31)所示，序列 $x[n]$的离散时间傅里叶变换 $X(e^{j\omega})$，可以用 MATLAB 函数 freqz 非常方便地在给定的 L 个离散频率点 $\omega=\omega_\ell$ 处进行计算。由于 $X(e^{j\omega})$是 ω 的连续函数，需要尽可能大地选取 L 的值，以使得命令 plot 产生的图形和真实离散傅里叶变换的图形尽可能一致。在 MATLAB 中，freqz 计算出序列$\{p_0\quad p_1\cdots p_M\}$和序列$\{d_0\quad d_1\cdots d_M\}$的 L 点离散傅里叶变换，然后对其离散傅里叶变换值相比得到 $X(e^{j\omega_\ell})$，$\ell=1,2,\cdots,L$。为更加方便快速地运算，应将 L 的值选为 2 的幂，如 256 或者 512。

程序 P3.1 可用来计算并画出式(3.31)给出的离散时间傅里叶变换。

```
% 程序 P3.1
% 离散时间傅里叶的求取
clf;
% 计算离散时间傅里叶变换的频率样本
w = -4*pi:8*pi/511:4*pi;
num = [2 1];den = [1 -0.6];
h = freqz(num,den,w);
% Plot the DTFT
subplot(2,1,1)
plot(w/pi,real(h));grid
title('H(e^{j\omega})的实部')
xlabel('\omega /\pi');
ylabel('振幅');
subplot(2,1,2)
plot(w/pi,imag(h));grid
title('H(e^{j\omega})的虚部')
xlabel('\omega /\pi');
ylabel('振幅');
pause
subplot(2,1,1)
plot(w/pi,abs(h));grid
title('|H(e^{j\omega})|幅度谱')
xlabel('\omega /\pi');
ylabel('振幅');
subplot(2,1,2)
plot(w/pi,angle(h));grid
title('相位谱 arg[H(e^{j\omega})]')
xlabel('\omega /\pi');
ylabel('以弧度为单位的相位');
```

习题

Q3.1 在程序P3.1中,计算离散时间傅里叶变换的原始序列是什么? MATLAB命令 pause 的作用是什么?

Q3.2 运行程序P3.1,求离散时间傅里叶变换的实部、虚部以及幅度和相位谱。离散时间傅里叶变换是 ω 的周期函数吗? 若是,周期是多少? 描述这四个图形表示的对称性。

Q3.3 修改程序P3.1,在范围 $0\leqslant\omega\leqslant\pi$ 内计算如下序列的离散时间傅里叶变换:

$$U(e^{j\omega})=\frac{0.7-0.5e^{-j\omega}+0.3e^{-j2\omega}+e^{-j3\omega}}{1+0.3e^{-j\omega}-0.5e^{-j2\omega}+0.7e^{-j3\omega}}$$

并重做习题Q3.2。讨论你的结果。你能解释相位谱中的跳变吗? MATLAB命令 unwrap 可以移除跳变。试求跳变被移除后的相位谱。

Q3.4 修改程序P3.1,计算如下有限长序列的离散时间傅里叶变换:

$$g[n]=[1\quad 3\quad 5\quad 7\quad 9\quad 11\quad 13\quad 15\quad 17]$$

并重做习题Q3.2。讨论你的结果。你能解释相位谱中的跳变吗?

Q3.5 怎样修改程序P3.1,才能使所画图形中的相位的单位是用度来衡量的?

例3.2 离散时间傅里叶变换的性质

大多数离散时间傅里叶变换的性质可以用MATLAB来验证。在本练习中,我们将验证R3.6中列出的性质。由于MATLAB中所有的数据都是有限长的向量,用来验证这些性质的序列因此都限制为有限长。

程序P3.2用于验证离散时间傅里叶变换的时移特性。

```
%程序 P3.2
%离散时间傅里叶变换的时移性质
clf;
w = -pi:2*pi/255:pi; wo = 0.4*pi; D = 10;
num = [1 2 3 4 5 6 7 8 9];
h1 = freqz(num,1,w);
h2 = freqz([zeros(1,D) num],1,w);
subplot(2,2,1)
plot(w/pi,abs(h1));grid
title('原序列的幅度谱')
subplot(2,2,2)
plot(w/pi,abs(h2));grid
title('时移后序列的幅度谱')
subplot(2,2,3)
plot(w/pi,angle(h1));grid
title('原序列的相位谱')
subplot(2,2,4)
plot(w/pi,angle(h2));grid
title('时移后序列的相位谱')
```

习题

Q3.6 通过加入合适的注释语句和程序语句,修改程序P3.2,对程序生成的图形中的两个轴加标记。哪个参数控制时移量?

Q3.7 运行修改后的程序并讨论你的结果。

Q3.8 选取不同的时移值重做习题 Q3.7。

Q3.9 选取两个改变了长度的序列以及两个不同的时移值，重做习题 Q3.7。

程序 P3.3 用于验证离散时间傅里叶变换的频移性质。

```
% 程序 P3.3
% 离散时间傅里叶变换的频移性质
clf;
w = -pi:2*pi/255:pi; wo = 0.4*pi;
num1 = [1 3 5 7 9 11 13 15 17];
L = length(num1);
h1 = freqz(num1,1,w);
n = 0:L-1;
num2 = exp(wo*i*n).*num1;
h2 = freqz(num2,1,w);
subplot(2,2,1)
plot(w/pi,abs(h1));grid
title('原序列的幅度谱')
subplot(2,2,2)
plot(w/pi,abs(h2));grid
title('频移后序列的幅度谱')
subplot(2,2,3)
plot(w/pi,angle(h1));grid
title('原序列的相位谱')
subplot(2,2,4)
plot(w/pi,angle(h2));grid
title('频移后序列的相位谱')
```

习题

Q3.10 通过加入合适的注释语句和程序语句，修改程序 P3.3，对程序生成的图形中的两个轴加标记。哪个参数控制频移量？

Q3.11 运行修改后的程序并讨论你的结果。

Q3.12 选取不同的频移值，重做习题 Q3.11。

Q3.13 选取两个改变了长度的序列以及两个不同的频移值，重做习题 Q3.11。

程序 P3.4 用于验证离散时间傅里叶变换的卷积性质。

```
% 程序 P3.4
% 离散傅里叶变换的卷积性质
clf;
w = -pi:2*pi/255:pi;
x1 = [1 3 5 7 9 11 13 15 17];
x2 = [1 -2 3 -2 1];
y = conv(x1,x2);
h1 = freqz(x1,1,w);
h2 = freqz(x2,1,w);
hp = h1.*h2;
h3 = freqz(y,1,w);
subplot(2,2,1)
```

```
plot(w/pi,abs(hp));grid
title('幅度谱的乘积')
subplot(2,2,2)
plot(w/pi,abs(h3));grid
title('卷积后序列的幅度谱')
subplot(2,2,3)
plot(w/pi,angle(hp));grid
title('相位谱的和')
subplot(2,2,4)
plot(w/pi,angle(h3));grid
title('卷积后序列的相位谱')
```

习题

Q3.14 通过加入合适的注释语句和程序语句,修改程序 P3.4,对程序生成的图形中的两个轴加标记。

Q3.15 运行修改后的程序并讨论你的结果。

Q3.16 选取两个改变了长度的序列,重做习题 Q3.15。

程序 P3.5 用于验证离散时间傅里叶变换的调制性质。

```
% 程序 P3.5
% 离散傅里叶变换的调制性质
clf;
w = -pi:2*pi/255:pi;
x1 = [1 3 5 7 9 11 13 15 17];
x2 = [1 -1 1 -1 1 -1 1 -1 1];
y = x1.*x2;
h1 = freqz(x1,1,w);
h2 = freqz(x2,1,w);
h3 = freqz(y,1,w);
subplot(3,1,1)
plot(w/pi,abs(h1));grid
title('第一个序列的幅度谱')
subplot(3,1,2)
plot(w/pi,abs(h2));grid
title('第二个序列的幅度谱')
subplot(3,1,3)
plot(w/pi,abs(h3));grid
title('乘积序列的幅度谱')
```

习题

Q3.17 通过加入合适的注释语句和程序语句,修改程序 P3.5,对程序生成的图形中的两个轴加标记。

Q3.18 运行修改后的程序并讨论你的结果。

Q3.19 选取两个改变了长度的序列,重做习题 Q3.18。

程序 P3.6 用于验证离散傅里叶变换的时间反转性质。

```
% 程序 P3.6
% 离散傅里叶变换的时间反转性质
```

```
clf;
w = -pi:2*pi/255:pi;
num = [1 2 3 4];
L = length(num) - 1;
h1 = freqz(num,1,w);
h2 = freqz(fliplr(num),1,w);
h3 = exp(w*L*i).*h2;
subplot(2,2,1)
plot(w/pi,abs(h1));grid
title('原序列的幅度谱')
subplot(2,2,2)
plot(w/pi,abs(h3));grid
title('时间反转后序列的幅度谱')
subplot(2,2,3)
plot(w/pi,angle(h1));grid
title('原序列的相位谱')
subplot(2,2,4)
plot(w/pi,angle(h3));grid
title('时间反转后序列的相位谱')
```

习题

Q3.20 通过加入合适的注释语句和程序语句,修改程序P3.6,对程序生成的图形中的两个轴加标记。试解释程序怎样进行时间反转运算。

Q3.21 运行修改后的程序并讨论你的结果。

Q3.22 选取两个不同长度的序列,重做习题Q3.21。

3.5 离散傅里叶变换

在MATLAB中,使用函数 fft 可以很容易地计算有限长序列 $x[n]$ 的离散傅里叶变换 $X[k]$。此函数有两种形式,fft(x)计算序列 $x[n]$ 的离散傅里叶变换值 $X[k]$,这里 $X[k]$ 的长度与 $x[n]$ 的长度相等。Fft(x,L)计算序列 $x[n]$ 的 L 点离散傅里叶变换,其中 $L \geqslant N$。若 $L > N$,在计算离散傅里叶变换之前,对 $x[n]$ 尾部的 $L - N$ 个值进行补零。同样,离散傅里叶变换序列 $X[k]$ 的离散傅里叶逆变换 $x[n]$ 用函数 ifft 计算,它也具有两种形式。

例3.3 离散傅里叶变换和离散傅里叶逆变换的计算

习题

Q3.23 编写一个MATLAB程序,计算并画出长度为 N 的 L 点离散傅里叶变换 $X[k]$ 的值,其中 $L \geqslant N$,然后计算并画出 L 点离散傅里叶逆变换 $X[k]$。对不同长度 N 和不同的离散傅里叶变换长度 L,运行程序。讨论你的结果。

Q3.24 编写一个MATLAB程序,用一个 N 点复数离散傅里叶变换计算两个长度为 N 的实数序列的 N 点离散傅里叶变换,并将结果同直接使用两个N点离散傅里叶变换得到的结果进行比较(见R3.13)。

Q3.25 编写一个MATLAB程序,用两个 N 点离散傅里叶变换计算一个长度为 $2N$ 的实数序列的 $2N$ 点离散傅里叶变换,并将结果同直接使用一个 $2N$ 点离散傅里叶变换得到的结果进行比较(见R3.14)。

例 3.4 离散傅里叶变换的性质

在离散傅里叶变换的应用中,相同长度序列的圆周移位和圆周卷积是两个非常重要的概念。在验证离散傅里叶变换的某些性质时,我们需要用到这些运算。下面我们用 MATLAB 函数 circshift 和 circonv 来实现它们。

```
function y = circshift(x,M)
%通过对一个有限长
%序列圆周移位 M 个样本
%得到一个序列 y
if abs(M) > length(x)
    M = rem(M,length(x));
end
if M < 0
    M = M + length(x);
end
y = [x(M+1:length(x)) x(1:M)];

function y = circonv(x1,x2)
L1 = length(x1); L2 = length(x2);
if L1 ~= L2,error('长度不相等的序列'),end
y = zeros(1,L1);
x2tr = [x2(1) x2(L2:-1:2)];
for k = 1:L1
    sh = circshift(x2tr,1-k);
    h = x1.*sh;
    y(k) = sum(h);
end
```

习题

Q3.26 在函数 circshift 中,命令 rem 的作用是什么?

Q3.27 解释函数 circshift 怎样实现圆周移位运算。

Q3.28 在函数 circonv 中,运算符~=的作用是什么?

Q3.29 解释函数 circonv 怎样实现圆周卷积运算。

程序 P3.7 用于说明一个有限长序列圆周移位的概念。它使用函数 circshift。

```
%程序 P3.7
%一个序列圆周移位的说明
clf;
M = 6;
a = [0 1 2 3 4 5 6 7 8 9];
b = circshift(a,M);
L = length(a)-1;
n = 0:L;
subplot(2,1,1);
stem(n,a);axis([0,L,min(a),max(a)]);
title('原序列');
subplot(2,1,2);
```

```
stem(n,b);axis([0,L,min(a),max(a)]);
title(['圆周移位',num2str(M),'个样本得到的序列']);
```

习题

Q3.30 通过加入合适的注释语句和程序语句,修改程序 P3.7,对程序生成的图形中的两个轴加标记。哪一个参数决定时移量?若时移量大于序列长度,将会发生什么?

Q3.31 运行修改后的程序并验证圆周时移运算。

程序 P3.8 用于说明离散傅里叶变换的圆周时移性质。它使用函数 circshift。

```
% 程序 P3.8
% 离散傅里叶变换的圆周时移性质
clf;
x = [0 2 4 6 8 10 12 14 16];
N = length(x) - 1; n = 0:N;
y = circshift(x,5);
XF = fft(x);
YF = fft(y);
subplot(2,2,1)
stem(n,abs(XF));grid
title('原序列的离散傅里叶变换的幅度');
subplot(2,2,2)
stem(n,abs(YF));grid
title('圆周移位后的序列的离散傅里叶变换的幅度');
subplot(2,2,3)
stem(n,angle(XF));grid
title('原序列的离散傅里叶变换的相位');
subplot(2,2,4)
stem(n,angle(YF));grid
title('圆周移位后的序列的离散傅里叶变换的相位');
```

习题

Q3.32 通过加入合适的注释语句和程序语句,修改程序 P3.8,对程序生成的图形中的两个轴加标记。时移量是多少?

Q3.33 运行修改后的程序并验证离散傅里叶变换的圆周时移性质。

Q3.34 选取两个不同的时移量,重做习题 Q3.33。

Q3.35 选取两个不同长度的序列,重做习题 Q3.33。

程序 P3.9 用于说明离散傅里叶变换的圆周卷积性质。它使用函数 circonv。

```
% 程序 P3.9
% 离散傅里叶变换的圆周卷积性质
g1 = [1 2 3 4 5 6]; g2 = [1 -2 3 3 -2 1];
ycir = circonv(g1,g2);
disp('圆周卷积的结果 = ');disp(ycir)
G1 = fft(g1); G2 = fft(g2);
yc = real(ifft(G1.*G2));
disp('离散傅里叶变换乘积的离散傅里叶逆变换的结果 = ');disp(yc)
```

习题

Q3.36 运行程序 P3.9 并验证离散傅里叶变换的圆周卷积性质。

Q3.37 选取另外两组等长序列重做习题 Q3.36。

程序 P3.10 用于说明圆周卷积和线性卷积之间的关系(见 R3.10)。

```
% 程序 P3.10
% 通过圆周卷积的线性卷积
g1 = [1 2 3 4 5];g2 = [2 2 0 1 1];
g1e = [g1 zeros(1,length(g2) - 1)];
g2e = [g2 zeros(1,length(g1) - 1)];
ylin = circonv(g1e,g2e);
disp('通过圆周卷积的线性卷积 = ');disp(ylin);
y = conv(g1,g2);
disp('直接线性卷积 = ');disp(y)
```

习题

Q3.38 运行程序 P3.10 并验证线性卷积可通过圆周卷积得到。

Q3.39 选取两组长度不等的序列重做习题 Q3.38。

Q3.40 编写一个 MATLAB 程序,对两个序列做离散傅里叶变换,以生成它们的线性卷积。用此程序验证习题 Q3.38 和习题 Q3.39 的结果。

程序 P3.11 用于验证一个实周期序列的偶数部分的离散傅里叶变换、奇数部分的离散傅里叶变换以及该序列自身的离散傅里叶变换之间的关系(见 R3.12)。

```
% 程序 P3.11
% 一个实序列的周期偶部分和周期奇部分
% 的离散傅里叶变换之间的关系
x = [1 2 4 2 6 32 6 4 2 zeros(1,247)];
x1 = [x(1) x(256: - 1:2)];
xe = 0.5 * (x + x1);
XF = fft(x);
XEF = fft(xe);
clf;
k = 0:255;
subplot(2,2,1);
plot(k/128,real(XF)); grid;
ylabel('振幅');
title('Re(DFT\{x[n]\})');
subplot(2,2,2);
plot(k/128,imag(XF)); grid;
ylabel('振幅');
title('Im(DFT\{x[n]\})');
subplot(2,2,3);
plot(k/128,real(XEF)); grid;
xlabel('时间序号 n');ylabel('振幅');
title('Re(DFT\{x_{e}[n]\})');
subplot(2,2,4);
```

```
plot(k/128,imag(XEF)); grid;
xlabel('时间序号 n');ylabel('振幅');
title('Im(DFT\{x_{e}[n]\})');
```

习题

Q3.41　序列 x1[n]和 x[n]之间的关系是什么?

Q3.42　运行程序 P3.11。由于周期序列的偶数部分的离散傅里叶变换是原序列的 XEF 的实数部分,XEF 的虚部应该为零。你能验证它们吗?你怎样解释仿真结果?

Q3.43　试修改源程序,验证周期序列的偶数部分的离散傅里叶变换和 XEF 的虚部之间的关系。

帕斯瓦尔关系[见式(3.15)]可用下面的程序来验证。

```
% 程序 P3.12
% 帕斯瓦尔关系
x = [(1:128) (128: - 1:1)];
XF = fft(x);
a = sum(x. * x)
b = round(sum(abs(XF).^2)/256)
```

习题

Q3.44　运行程序 P3.12。你得到的 a 和 b 的值相等吗?

Q3.45　不使用命令 abs(XF)来修改程序。用 MATLAB 命令 conj(x)来计算 x 的复共轭。

3.6　z 变换

与离散傅里叶变换一样,这里我们仅限于讨论序列 $g[n]$的 z 变换 $G(z)$,它是复变量 z^{-1} 的有理函数,表示为式(3.24)中 z^{-1}的多项式之比,或式(3.25)中的因式形式。在实际中,有意义的运算如下。(1)在单位圆上求 z 变换 $G(z)$,即求 $G(e^{j\omega})$;(2)生成 $G(z)$的极零点图;(3)生成 $G(z)$的因式形式;(4)确定 $G(z)$的逆 z 变换 $g[n]$;(5) $G(z)$的部分分式展开。在接下来的两个练习中,我们将学习怎样使用 MATLAB 来进行上面的操作。

例 3.5　z 变换分析

函数 freqz 用来求单位圆上的有理 z 变换的值。因此,程序 P3.1 可以在不做任何修改的情况下直接使用。

习题

Q3.46　使用程序 P3.1 在单位圆上求下面的 z 变换:

$$G(z)=\frac{2+5z^{-1}+9z^{-2}+5z^{-3}+3z^{-4}}{5+45z^{-1}+2z^{-2}+z^{-3}+z^{-4}} \tag{3.32}$$

使用函数 zplane 可以很容易地得到有理 z 变换 $G(z)$的极零点图。该函数有两种形式。若 z 变换用式(3.32)所示的有理函数的形式给出,使用的命令是 zplane(num, den),其中 num 和 den 是按 z^{-1}的升幂排列的、$G(z)$的分子和分母多项式系数的行向

量。另外,若给出 $G(z)$的零点和极点,将用到命令 zplane(zeros,poles),其中 zeros 和 poles 都是列向量。在由 MATLAB 产生的极零点图中,极点的位置用符号 × 表示,而零点的位置用符号○表示。

函数 tf2zp 用来确定有理 z 变换 $G(z)$的零点和极点。用到的程序语句是[z,p,k] = tf2zp(num,den),其中 num 和 den 是按 z^{-1}的升幂排列的、$G(z)$的分子和分母多项式系数的行向量,而输出包含增益常数 k,以及分别以列向量 z 和 p 表示出的零点和极点。使用函数 sos = zp2sos(z,p,k)可以将零极点形式的 z 变换表示成因式形式。函数计算出以因式形式表示的二阶因子的系数,$L\times6$ 矩阵 sos 得到了每个二阶因式的系数,矩阵 sos 为

$$\text{sos}=\begin{bmatrix} b_{01} & b_{11} & b_{21} & a_{01} & a_{11} & a_{21}\\ b_{02} & b_{12} & b_{22} & a_{02} & a_{12} & a_{22}\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ b_{0L} & b_{1L} & b_{2L} & a_{0L} & a_{1L} & a_{2L} \end{bmatrix}$$

式中第 ℓ 行表示的是第ℓ 个 z 变换 $G(z)$的二阶因式的分子和分母的系数。

$$G(z)=\prod_{\ell=1}^{L}\frac{b_{0\ell}+b_{1\ell}z^{-1}+b_{2\ell}z^{-2}}{a_{0\ell}+a_{1\ell}z^{-1}+a_{2\ell}z^{-2}}$$

习题

Q3.47 编写一个 MATLAB 程序,计算并显示零点和极点,计算并显示其因式形式,并产生以 z^{-1} 的两个多项式之比的形式表示的 z 变换的极零点图。使用该程序,分析式(3.32)的 z 变换 $G(z)$。

Q3.48 通过习题 Q3.47 产生的极零点图,求出 $G(z)$的收敛域的数目。清楚地显示所有的收敛域。由极零点图,说明离散时间傅里叶变换是否存在。

将零点、极点和增益常量转换为有理形式的 z 变换转换的逆过程,可以用函数 zp2tf 实现。用到的程序语句是[num,den] = zp2tf(z,p,k)。

习题

Q3.49 编写一个 MATLAB 程序,通过零点、极点和增益常量计算并显示有理 z 变换。使用这个程序确定有如下参数的 z 变换有理式:零点为 $\xi_1=0.3$,$\xi_2=2.5$,$\xi_3=-0.2+j0.4$,$\xi_4=-0.2-j0.4$,极点为 $\lambda_1=0.5$,$\lambda_2=-0.75$,$\lambda_3=0.6+j0.7$,$\lambda_4=0.6-j0.7$,增益常数为 $k=3.9$。

例 3.6 逆 z 变换

求 $G(z)$的逆 z 变换 $g[n]$时,可以用 MATLAB 以两种不同的方式计算。为此,需要预先知道 $G(z)$的收敛域。函数 impz 提供了时域序列的样本,该序列被假定为因果的。这个函数有三种不同的形式:[g,t] = impz(num,den),[g,t] = impz(num,den,L),[g,t] = impz(num,den,L,FT)。其中 num 和 den 是按 z^{-1}的升幂排列的、分子和分母多项式系数的行向量,L 是所求逆变换的样本数,g 是包含从样本 n = 0 开始的逆变换的样本向量,t 是 g 的长度,且 FT 是单位为 Hz 的给定抽样频率,其默认值为 1。

有理逆 z 变换的完整形式可先通过函数 residuez 求出部分分式展开,接着通过查 z 变换表来确定展开式中每一项的逆 z 变换。函数 residuez 也可以用来把以部分分式展开式形式给出的 z 变换表达式转换成 z^{-1} 的多项式之比。

习题

Q3.50 编写一个 MATLAB 程序,计算一个有理逆 z 变换的前 L 个样本,其中 L 的值由用户通过命令 input 提供。用该程序计算并画出式(3.32)中 $G(z)$ 的逆变换的前 50 个样本。使用命令 stem 画出由逆变换产生的序列。

Q3.51 编写一个 MATLAB 程序,确定一个有理 z 变换的部分分式展开。用该程序求出式(3.32)中 $G(z)$ 的部分分式表达式和以完整形式表示的其逆 z 变换 $g[n]$。假定 $g[n]$ 是一个因果序列。

3.7 基础知识阅读

[1] A. Antoniou. *Digital Filters: Analysis, Design, and Applications*. McGraw-Hill, New York NY, 1993, second edition. Chs. 2, 13.

[2] E. Cunningham. *Digital Filtering: An Introduction*. Houghton-Mifflin, Boston, MA, 1992. Ch. 3.

[3] D.J. DeFatta, J.G. Lucas, and W.S. Hodgkiss. *Digital Signal Processing: A System Design Approach*. Wiley, New York, NY, 1988. Secs. 2.1, 3.1–3.3, 6.1–6.4.

[4] L.B. Jackson, *Digital Filters and Signal Processing*. Kluwer, Boston MA, third edition, 1996. Ch. 3 and Secs. 6.1, 6.2, 7.1, 7.2.

[5] R. Kuc. *Introduction to Digital Signal Processing*. McGraw-Hill, New York NY, 1988. Chs. 3–5.

[6] L.C. Ludeman. *Fundamentals of Digital Signal Processing*. Harper & Row, New York NY, 1986. Secs. 1.4, 2.1, 2.2, 6.3.

[7] S.K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, New York NY, 1998. Ch. 3.

[8] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs NJ, 1989. Secs. 2.6–2.9 and Chs. 4, 8.

[9] S.J. Orfanidis. *Introduction to Signal Processing*. Prentice-Hall, Englewood Cliffs NJ, 1996. Ch. 5 and Secs. 9.1, 9.2.

[10] B. Porat. *A Course in Digital Signal Procesing*. Wiley, New York NY, 1996. Secs. 2.7, 4.1–4.7, 7.1–7.6.

[11] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs NJ, third edition, 1996. Secs. 1.3, 3.1–3.4, 4.2, 4.3, 5.1, 5.2.

[12] R.A. Roberts and C.T. Mullis. *Digital Signal Processing*. Addison-Wesley, Reading MA, 1987. Chs. 3, 4.

第 4 章　线性时不变离散时间系统的频域分析

4.1　导言

在时域中,我们用系统的冲激响应来描述线性时不变(LTI)离散时间系统。在任意输入情况下,将输入序列与系统的冲激响应做卷积,可得系统的输出响应。对某些线性时不变离散时间系统,也可以用线性常系数差分方程来进行描述。对这些系统,在任意输入情况下,输出序列可进行递归计算。若对卷积或差分方程描述的系统应用离散时间傅里叶变换或 z 变换,则线性时不变离散时间系统也可以在频域中进行描述。除了使系统的设计和实现在某些应用下更加简单外,这种变换域表示方法还能观察出系统的某些特性。

4.2　基础知识回顾

R4.1 若$\{h[n]\}$表示一个线性时不变离散时间系统的冲激响应,对$\{h[n]\}$做离散时间傅里叶变换得到其*频率响应* $H(e^{j\omega})$,即

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h[n]\, e^{-j\omega n} \tag{4.1}$$

R4.2 通常,$H(e^{j\omega})$是一个周期为 2π 的 ω 的复值函数,可以根据实部、虚部或者幅度相位来表示该函数。因此,

$$H(e^{j\omega}) = H_{re}(e^{j\omega}) + j\, H_{im}(e^{j\omega}) = |H(e^{j\omega})|\, e^{j\,\theta(\omega)} \tag{4.2}$$

其中 $H_{re}(e^{j\omega})$和 $H_{im}(e^{j\omega})$分别是 $H(e^{j\omega})$的实数和虚部,并且

$$\theta(\omega) = arg\{H(e^{j\omega})\} \tag{4.3}$$

$|H(e^{j\omega})|$称为*幅度响应*,而 $\theta(\omega)$称为线性时不变离散时间系统的*相位响应*。

R4.3 线性时不变系统的*增益函数*$\mathcal{G}(\omega)$定义为

$$\mathcal{G}(\omega) = 20\log_{10}|H(e^{j\omega})| \quad \text{dB} \tag{4.4}$$

增益函数的相反数 $a(\omega) = -\mathcal{G}(\omega)$,称为*衰减*或*损益函数*。

R4.4 对于用实冲激响应 $h[n]$描述的离散时间系统,幅度函数是 ω 的偶函数,即$|H(e^{j\omega})| = |H(e^{-j\omega})|$;而相位函数是 ω 的奇函数,即 $\theta(\omega) = -\theta(-\omega)$。同样,$H_{re}(e^{j\omega})$是 ω 的偶函数,而 $H_{im}(e^{j\omega})$是 ω 的奇函数。

R4.5 离散时间系统的相位响应在计算机中计算时,可能会因计算反正切函数而引起 2π 数值的相位跳变。在跳变点增加 $\pm 2\pi$ 的倍数,可以使系统的相位响应成为 ω 的连续函数。

R4.6 线性时不变(LTI)离散时间系统的*群延迟函数*定义为

$$\tau(\omega) = -\frac{d\theta_c(\omega)}{d\omega} \tag{4.5}$$

其中 $\theta_c(\omega)$表示展开的相位函数。若相位函数的单位是弧度，那么群延迟的单位是秒。

R4.7　在输入

$$x[n] = A\cos(\omega_o n + \phi) \tag{4.6}$$

(其中 A 是实数)的情况下，频率响应为 $H(e^{j\omega})$的实系数线性时不变离散时间系统的稳态输出 $y[n]$为

$$y[n] = A\,|H(e^{j\omega_o})|\cos(\omega_o n + \theta(\omega_o) + \phi) \tag{4.7}$$

R4.8　由式(2.4)描述的线性时不变离散时间系统的卷积可知，线性时不变离散时间系统的频率响应可以由输出序列 $y[n]$的傅里叶变换 $Y(e^{j\omega})$与输入序列 $x[n]$的傅里叶变换 $X(e^{j\omega})$相比得到，即

$$H(e^{j\omega}) = Y(e^{j\omega})/X(e^{j\omega}) \tag{4.8}$$

R4.9　对用形如式(2.11)所示线性常系数差分方程描述的线性时不变系统，频率响应 $H(e^{j\omega})$可表示为

$$H(e^{j\omega}) = \frac{\sum_{k=0}^{M} p_k e^{-j\omega k}}{\sum_{k=0}^{N} d_k e^{-j\omega k}} \tag{4.9}$$

R4.10　线性时不变离散时间系统的冲激响应$\{h[n]\}$的 z 变换 $H(z)$，称为传输函数或系统函数。由式(2.4)描述的线性时不变离散时间系统的卷积可知，线性时不变离散时间系统的传输函数 $H(z)$由输出序列 $y[n]$的 z 变换 $Y(z)$与输入序列 $x[n]$的 z 变换 $X(z)$相比得到；即 $H(z) = Y(z)/X(z)$。

R4.11　若 $H(z)$的收敛域包括单位圆，则它与线性时不变离散时间系统的频率响应 $H(e^{j\omega})$的关系如下：

$$H(e^{j\omega}) = H(z)|_{z=e^{j\omega}} \tag{4.10}$$

R4.12　对实系数传输函数 $H(z)$，有

$$|H(e^{j\omega})|^2 = H(e^{j\omega})H^*(e^{j\omega}) = H(e^{j\omega})H(e^{-j\omega}) = H(z)H(z^{-1})|_{z=e^{j\omega}} \tag{4.11}$$

R4.13　对于式(2.11)中用线性常系数差分方程描述的线性时不变系统，传输函数 $H(z)$可以表示为

$$H(z) = \frac{Y(z)}{X(z)} = \frac{p_0 + p_1 z^{-1} + \cdots + p_M z^{-M}}{d_0 + d_1 z^{-1} + \cdots + d_N z^{-N}} \tag{4.12}$$

R4.14　式(4.12)给出的传输函数也可用下面的形式表示：

$$H(z) = \frac{p_0}{d_0}\frac{\prod_{k=1}^{M}(1 - \xi_k z^{-1})}{\prod_{k=1}^{N}(1 - \lambda_k z^{-1})} \tag{4.13}$$

其中 $\xi_1, \xi_2, \cdots, \xi_M$ 是 $H(z)$的 M 个零点，而 $\lambda_1, \lambda_2, \cdots, \lambda_N$ 是 $H(z)$的 N 个极点。若 $N > M$，在 $z = 0$ 处有另外$(N - M)$个零点，而若 $N < M$，在 $z = 0$ 处有另外$(M - N)$个极点。

R4.15　稳定因果系统的传输函数 $H(z)$的所有极点，都必须严格在单位圆内。

R4.16 图 4.1 中给出了四种常见的理想零相位数字滤波器的频率响应,这些数字滤波器都具有实冲激响应系数。理想滤波器的幅度响应在通带中等于 1,而在阻带中等于零,并且在任何频率上相位都为零。

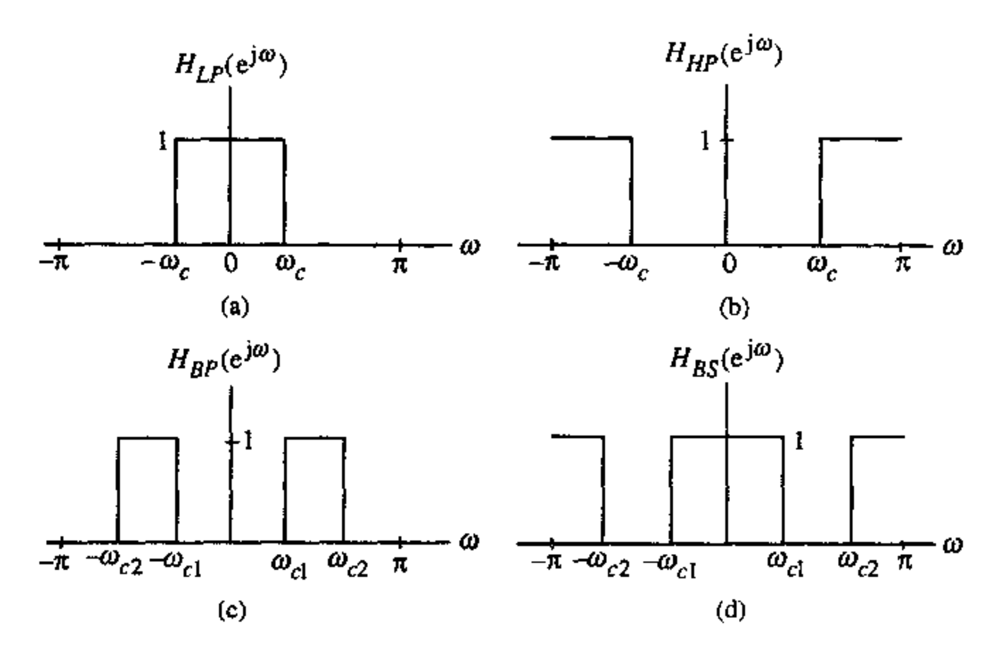


图 4.1 理想滤波器的频率响应:(a)低通滤波器;(b)高通滤波器;(c)带通滤波器;(d)带阻滤波器

R4.17 图 4.1 所示的理想低通滤波器的冲激响应 $h_{LP}[n]$ 为

$$h_{LP}[n]=\frac{\sin(\omega_c n)}{\pi n},\quad -\infty<n<\infty \tag{4.14}$$

R4.18 一阶低通无限冲激响应传输函数 $H_{LP}(z)$ 为

$$H_{LP}(z)=\frac{1-\alpha}{2}\cdot\frac{1+z^{-1}}{1-\alpha z^{-1}} \tag{4.15}$$

其中为了稳定性,令 $|\alpha|<1$。比 dc($\omega=0$)处的最大值小 3 dB 的频率点 ω_c,称为 3 dB 截止频率,该值与参数 α 的关系如下:

$$\alpha=\frac{1-\sin\omega_c}{\cos\omega_c} \tag{4.16}$$

R4.19 一阶高通无限冲激响应系统的传输函数 $H_{HP}(z)$ 为

$$H_{HP}(z)=\frac{1+\alpha}{2}\cdot\frac{1-z^{-1}}{1-\alpha z^{-1}} \tag{4.17}$$

其中为了稳定性,令 $|\alpha|<1$。其 3 dB 截止频率 ω_c 同样由式(4.16)给出。

R4.20 二阶带通无限冲激响应系统的传输函数 $H_{BP}(z)$ 为

$$H_{BP}(z)=\frac{1-\alpha}{2}\cdot\frac{1-z^{-2}}{1-\beta(1+\alpha)z^{-1}+\alpha z^{-2}} \tag{4.18}$$

其幅度响应在 $\omega=0$ 和 $\omega=\pi$ 处趋近于零,而在 $\omega=\omega_o$ 处取最大值 1,ω_o 称为带通滤波器的中心频率,其中

$$\omega_o = \cos^{-1}(\beta) \tag{4.19}$$

由两个 3 dB 截止频率的差值得到的 3 dB 带宽 $\Delta\omega_{3dB}$ 为

$$\Delta\omega_{3dB} = \omega_{c2} - \omega_{c1} = \cos^{-1}\left(\frac{2\alpha}{1+\alpha^2}\right) \tag{4.20}$$

R4.21 二阶带阻无限冲激响应系统的传输函数 $H_{BS}(z)$ 为

$$H_{BS}(z) = \frac{1+\alpha}{2}\cdot\frac{1-2\beta z^{-1}+z^{-2}}{1-\beta(1+\alpha)z^{-1}+\alpha z^{-2}} \tag{4.21}$$

其幅度响应在 $\omega=0$ 和 $\omega=\pi$ 处取最大值 1，而在 $\omega=\omega_o$ 处趋近于零，ω_o 称为陷波频率，其中 ω_o 由式(4.19)给出。3 dB 陷波带宽 $\Delta\omega_{3dB}$ 由式(4.20)给出。

R4.22 通过级联上面提到的简单数字滤波器，可得到具有更尖锐幅度响应的数字滤波器。例如，将 K 个由式(4.15)描述的一阶低通部分级联，得到的整个结构为传输函数 $G_{LP}(z)$

$$G_{LP}(z) = \left(\frac{1-\alpha}{2}\cdot\frac{1+z^{-1}}{1-\alpha z^{-1}}\right)^K \tag{4.22}$$

参数 α 和 K 与级联的 3 dB 截止频率 ω_c 的关系如下所示：

$$\alpha = \frac{1+(1-B)\cos\omega_c - \sin\omega_c\sqrt{2B-B^2}}{1-B+\cos\omega_c} \tag{4.23}$$

其中，

$$B = 2^{(K-1)/K} \tag{4.24}$$

R4.23 对有限长实输入信号的非实时处理，若对因果性要求不太严格，非常容易实现零相位滤波。如图 4.2 所示，在一个系统中，有限长输入数据通过因果实系数滤波器 $H(z)$ 处理，对该滤波器的输出进行时间反转，再通过同样的滤波器处理，就可以实现零相位滤波。

$x[n]$ → $H(z)$ → $v[n]$　　$u[n]$ → $H(z)$ → $w[n]$

$u[n]=v[-n],\quad y[n]=w[-n]$

图 4.2　一种零相位滤波方案的实现

R4.24 通常我们能设计出具有精确线性相位响应的有限冲激响应系统。该系统的传输函数符合由

$$h[n] = h[N-n],\qquad 0\leqslant n\leqslant N \tag{4.25}$$

定义的对称冲激响应，或者由

$$h[n] = -h[N-n],\qquad 0\leqslant n\leqslant N \tag{4.26}$$

定义的反对称冲激响应，其中 N 是传输函数的阶数，$h[n]$ 的长度是 $N+1$。这样的传输函数有四种类型：

1 型：奇数长度对称冲激响应。

2 型：偶数长度对称冲激响应。

3 型:奇数长度反对称冲激响应。

4 型:偶数长度反对称冲激响应。

R4.25 2 型有限冲激响应系统的传输函数在 $z=-1$ 处有一个零点,因此,它不能用来设计高通滤波器。3 型有限冲激响应系统的传输函数在 $z=1$ 和 $z=-1$ 处各有一个零点,因此不能用于设计低通、高通或带通滤波器。由于在 $z=1$ 处存在一个零点,4 型有限冲激响应系统的传输函数不适合设计低通滤波器。1 型有限冲激响应滤波器没有这种限制,因此可用于设计任何类型的滤波器。

R4.26 若满足

$$|H(e^{j\omega})|\leqslant 1 \quad \text{对于所有 } \omega \tag{4.27}$$

则因果稳定实系数系统的传输函数 $H(z)$定义为有界实(BR)传输函数。

R4.27 在所有频率下具有单位幅度响应的传输函数 $A(z)$,

$$|A(e^{j\omega})|^2=1 \quad \text{对于所有 } \omega \tag{4.28}$$

称为全通传输函数。M 阶因果实系数无限冲激响应系统的全通传输函数如下所示:

$$A(z)=\pm\frac{z^{-M}D_M(z^{-1})}{D_M(z)} \tag{4.29}$$

其中 $D_M(z)$是最高阶数为 M 的多项式。实系数全通系统的传输函数的极点和零点在 z 平面上表现出镜像对称。若全通系统的传输函数也是因果和稳定的,则它的所有极点在单位圆内,它的所有零点在单位圆外并且与极点镜像对称。

R4.28 若因果稳定系统的传输函数的所有零点在单位圆内或单位圆上,则该系统的传输函数称为最小相位系统的传输函数,而若因果稳定系统的传输函数的所有零点在单位圆外,则该系统的传输函数称为最大相位系统的传输函数。

R4.29 若一组 M 个系统的传输函数的总和等于单位延时的某个整倍数,即

$$\sum_{k=0}^{M-1}H_k(z)=\beta z^{-n_o}, \qquad \beta\neq 0 \tag{4.30}$$

则该 M 个传输函数$\{H_0(z),H_1(z),\cdots,H_{M-1}(z)\}$称为相互延时互补,其中 n_o 是非负整数。奇数长度 L 的 1 型线性相位有限冲激响应系统的传输函数 $H_0(z)$的延时互补传输函数 $H_1(z)$,可以由式 $H_1(z)=z^{-(L-1)/2}-H_0(z)$得到。

R4.30 若一组 M 个系统的传输函数的总和等于全通系统的函数 $A(z)$,即

$$\sum_{i=0}^{M-1}H_i(z)=A(z) \tag{4.31}$$

则该 M 个数字滤波器$\{H_i(z)\}$, $i=0,1,\cdots,M-1$ 称为相互全通互补。

R4.31 若一组 M 个系统的幅度响应的平方和等于 1,即

$$\sum_{i=0}^{M-1}|H_i(e^{j\omega})|^2=1 \quad \text{对于所有 } \omega \tag{4.32}$$

则该 M 个数字滤波器$\{H_i(z)\}$, $i=0,1,\cdots,M-1$ 称为相互功率互补。

R4.32　设 $A_m(z)$是一个 m 阶实系数全通系统的传输函数：

$$A_m(z)=\frac{d_m+d_{m-1}z^{-1}+d_{m-2}z^{-2}+\ldots+d_1z^{-(m-1)}+z^{-m}}{1+d_1z^{-1}+d_2z^{-2}+\ldots+d_{m-1}z^{-(m-1)}+d_mz^{-m}} \tag{4.33}$$

根据

$$\begin{aligned}A_{m-1}(z)&=z\left[\frac{A_m(z)-k_m}{1-k_mA_m(z)}\right]\\&=\frac{d'_{m-1}+d'_{m-2}z^{-1}+\ldots+d'_1z^{-(m-2)}+z^{-(m-1)}}{1+d'_1z^{-1}+\ldots+d'_{m-2}z^{-(m-2)}+d'_{m-1}z^{-(m-1)}}\end{aligned} \tag{4.34}$$

可产生$(m-1)$阶实系数全通系统的传输函数 $A_{m-1}(z)$。其中，

$$d'_i=\frac{d_i-d_md_{m-i}}{1-d_m^2},\qquad i=1,2,\cdots,m-1 \tag{4.35}$$

定义 $k_m=A_m(\infty)$。$A_m(z)$稳定的充要条件为：(1)$k_m^2<1$，(2)$A_{m-1}(z)$是稳定全通系统的传输函数。可通过产生$(m-2)$阶全通系统的传输函数来检测 $A_{m-1}(z)$的稳定性，依次类推，得到一组阶数递减的全通函数

$$A_m(z),\ A_{m-1}(z),\ \cdots,\ A_2(z),\ A_1(z),\ A_0(z)=1$$

和一组系数

$$k_m,\ k_{m-1},\ \cdots,\ k_2,\ k_1$$

当且仅当 $k_\ell^2<1$ 且满足 $\ell=m,m-1,\cdots,1$ 时，全通函数 $A_m(z)$是稳定的。

4.3　用到的 MATLAB 命令

这个练习中用到的 MATLAB 命令如下：

通用命令

```
disp
```

运算符和特殊符号

```
:    .    +    -    *    /    ;    %
```

语言构造与调试

```
function    pause
```

基本矩阵和矩阵控制

```
fliplr    pi
```

基本函数

```
abs    angle    imag    log10    real
```

二维图形

```
axis    grid    plot    stem    title    xlabel    ylabel
```

通用图形函数

```
clf    subplot
```

信号处理工具箱

```
filter    filtfilt    freqz    grpdelay    impz    poly2rc    sinc    zplane
```

关于这些命令的辅助说明,请参见 *MATLAB Reference Guide*[Mat94]和 *Signal Processing Toolbox User's Guide*[Mat96],也可在命令窗口键入 help *命令名称*。在附录 B 中可找到此处使用的 MATLAB 函数的简要解释。

4.4 传输函数和频率响应

线性时不变离散时间系统的冲激响应$\{h[n]\}$的 z 变换是其传输函数 $H(z)$。若 $H(z)$的收敛域包含单位圆,即系统为稳定系统,则在单位圆上 $z=e^{j\omega}$处计算的是系统的频率响应 $H(e^{j\omega})$。在该练习中,我们将学习因果稳定线性时不变离散时间系统的不同性质。内容将涉及由传输函数求频率响应,计算系统的群延迟,求出表示系统的差分方程,从传输函数确定冲激响应,从传输函数生成极零点图,以及由极零点图研究该线性时不变系统的稳定性。

例 4.1 传输函数分析

式(4.9)给出了一个线性时不变离散时间系统的频率响应,该系统用形如式(2.11)的差分方程描述。频率响应可用命令 freqz 在一组离散频率点非常方便地计算得到。事实上,程序 P3.1 就可用于这个目的,同前一个练习中习题 Q3.41 所做的相同。

习题

Q4.1 修改程序 P3.1,取三个不同的 M 值,当 $0\leqslant\omega\leqslant2\pi$ 时计算并画出式(2.13)所示滑动平均滤波器的幅度和相位谱。证明由幅度和相位谱表现出的对称类型。它表示了哪种类型的滤波器?你现在能解释习题 Q2.1 的结果吗?

修改后的程序 P3.1 也可用于计算并画出线性时不变离散时间系统的频率响应,该系统用形如式(4.12)的传输函数描述。

习题

Q4.2 使用修改后的程序 P3.1,计算并画出当 $0\leqslant\omega\leqslant\pi$ 时传输函数

$$H(z)=\frac{0.15\,(1-z^{-2})}{1-0.5\,z^{-1}+0.7\,z^{-2}} \tag{4.36}$$

的因果线性时不变离散时间系统的频率响应。它表示哪种类型的滤波器?

Q4.3 对下面的传输函数重做习题 Q4.2:

$$G(z)=\frac{0.15\,(1-z^{-2})}{0.7-0.5\,z^{-1}+z^{-2}} \tag{4.37}$$

式(4.36)和式(4.37)给出的两个滤波器之间的区别是什么?你将选择哪一个滤波器来滤波,为什么?

为了保持我们所需频带内的信号波形，线性时不变离散时间系统的群延迟在这些频带内近似为常数。函数 grpdelay 可以很容易地计算一个传输函数的群延迟。

习题

Q4.4 使用 MATLAB 计算并画出当 $0 \leqslant \omega \leqslant \pi$ 时因果线性时不变离散时间系统的群延迟。系统的传输函数为

$$H(z)=\frac{z^{-1}-1.2\,z^{-2}+z^{-3}}{1-1.3\,z^{-1}+1.04\,z^{-2}-0.222\,z^{-3}} \tag{4.38}$$

函数 impz 可用来计算因果线性时不变离散时间系统的冲激响应的开始部分。因此，可使用习题 Q3.50 中你编写的程序。

习题

Q4.5 使用习题 Q3.50 中编写的程序，分别计算并画出式(4.36)和式(4.37)确定的两个滤波器的冲激响应中的前 100 个样本。讨论你的结果。

传输函数的极零点图同样能分析线性时不变离散时间系统的性质。使用命令 zplane 可以很容易地得到系统的极零点图。

习题

Q4.6 使用 zplane 分别生成式(4.36)和式(4.37)确定的两个滤波器的极零点图。讨论你的结果。

4.5 传输函数的类型

数字信号处理的一个主要应用是对离散时间信号滤波，以移除不需要的成分。图 4.1 显示了四类理想滤波器的频率响应。这些滤波器都具有两个无限冲激响应，因此在实际上是不可能非常精确地实现的。但在实际应用中，近似实现这些滤波器是可行的。在这个练习中，我们将学习近似实现的理想滤波器的一些性质。在第 7 章中，我们将设计给定指标的滤波器。

例 4.2 滤波器

在图 4.1 中，式(4.14)给出的理想低通滤波器的冲激响应 $h_{LP}[n]$是无限的，因此实际上不可能实现。在实际应用中，将冲激响应截短到一个有限值，可以近似地实现该滤波器。然而，将冲激响应截短后滤波器表现为一个非因果的滤波器。将经过截短后的滤波器的冲激响应向右移动 $N/2$ 个样本，可得因果上的近似，如下所示：

$$\hat{h}_{LP}[n]=\frac{\sin\omega_c(n-N/2)}{\pi(n-N/2)},\qquad 0\leqslant n\leqslant N \tag{4.39}$$

滤波器的长度为 $N+1$。

下面这个使用函数 sinc 的程序，可用于计算上述近似滤波器的冲激响应。

```
% 程序 P4.1
% 截短的理想低通滤波器的冲激响应
```

```
clf;
fc = 0.25;
n = [ - 6.5:1:6.5];
y = 2 * fc * sinc(2 * fc * n);k = n + 6.5;
stem(k,y);title('N = 13');axis([0 13 -0.2 0.6]);
xlabel('时间序号 n');ylabel('振幅');grid
```

习题

Q4.7 用程序 P4.1 计算并画出近似理想低通滤波器的冲激响应。低通有限冲激响应滤波器的长度是多少？在程序 P4.1 中,哪个语句确定滤波器的长度？哪个参数控制截止频率？

Q4.8 修改程序 P4.1,计算并画出式(4.39)所示长度为 20、截止角频率为 $\omega_c = 0.45$ 的有限冲激响应低通滤波器的冲激响应。

Q4.9 修改程序 P4.1,计算并画出式(4.39)所示长度为 15、截止角频率为 $\omega_c = 0.65$ 的有限冲激响应低通滤波器的冲激响应。

Q4.10 编写一个 MATLAB 程序,计算并画出式(4.39)所示有限冲激响应低通滤波器的振幅响应。使用这个程序,选取几个不同的 N 值,画出振幅响应并讨论你的结果。

如习题 Q4.1 的结果所示,式(2.13)所示滑动平均滤波器也有一个低通幅度响应。最简单的这种滤波器的长度是 2,系统的传输函数为

$$H_0(z) = \frac{1}{2}(1 + z^{-1}) \tag{4.40}$$

可以看出,这种滤波器在 $\omega_c = \pi/2$ 处有一个 3 dB 截止频率。级联这些简单的低通滤波器,可得到具有更尖锐幅度响应的低通滤波器。将 $H_0(z)$的 K 个部分级联,在

$$\omega_c = 2\cos^{-1}(2^{-1/2K}) \tag{4.41}$$

处会有一个 3 dB 截止频率。稍微修改式(2.13)给出的差分方程,可产生一个高通滤波器,其传输函数为

$$H_1(z) = \frac{1}{M}\sum_{n=0}^{M-1}(-1)^n z^{-n} \tag{4.42}$$

下面给出的函数 gain 以 dB 为单位,计算并画出了一个有理传输函数的增益响应。

```
function [ g,w] = gain(num,den)
% 在单位圆上半部分的
% 256 个等空间点计算以 dB 为单位的
% 传输函数的增益响应
% 分子系数在向量 num 中
% 分母系数在向量 den 中
% 频率值以向量 w 返回
% 增益数值以向量 g 返回
w = 0:pi/255:pi;
h = freqz(num,den,w);
g = 20 * log10(abs(h));
```

程序 P4.2 说明了函数 gain 的使用,计算并画出了一个滑动平均低通滤波器的增益响应。

```
%程序 P4.2
%一个滑动平均低通滤波器的增益响应
clf;
M = 2;
num = ones(1,M)/M;
[g,w] = gain(num,1);
plot(w/pi,g);grid
axis([0 1 -50 0.5])
xlabel('\omega /\pi');ylabel('单位为 dB 的增益');
title(['M = ',num2str(M)])
```

习题

Q4.11 运行程序 P4.2,计算并画出一个长度为 2 的滑动平均滤波器的增益响应。从图中验证 3 dB 截止频率在 $\pi/2$ 处。

Q4.12 修改程序 P4.2,计算并画出 K 个长度为 2 的滑动平均滤波器级联后的增益响应。使用修改的程序,画出级联 3 个部分的滤波器的增益响应,并验证级联后的 3 dB 截止频率由式(4.41)给出。

Q4.13 修改程序 P4.2,计算并画出式(4.42)所示高通滤波器的增益响应。运行修改后的程序,画出 $M=5$ 时的增益响应,并从图中确定其 3 dB 截止频率。

在实际应用中,R4.18 到 R4.21 描述的简单一阶和二阶无限冲激响应滤波器已完全够用。若需要,这些滤波器中的任何一个都可以级联起来,从而提供更尖锐的增益响应。这些滤波器中的任何一个都具有下面的重要性质。显而易见,式(4.15)和式(4.17)中的 $H_{LP}(z)$ 和 $H_{HP}(z)$ 分别是全通互补和功率互补的。改变乘法器系数 α 的值,可以很容易地调节两个滤波器的 3 dB截止频率 ω_c。同样,式(4.18)和式(4.21)中的 $H_{BP}(z)$ 和 $H_{BS}(z)$ 也分别是全通互补和功率互补的。也可通过改变乘法器系数 β 的值来调节这两个滤波器的中心和陷波频率 ω_o,且调节乘法器系数 α 的值可以改变它们的 3 dB 带宽。

习题

Q4.14 设计一个在 0.45π 处具有 3 dB 截止角频率 ω_c 的一阶无限冲激响应低通滤波器和一阶无限冲激响应高通滤波器。用 MATLAB 计算并画出它们的增益响应,验证设计的滤波器是否满足指标。用 MATLAB 证明两个滤波器是全通互补和功率互补的。

Q4.15 级联 10 个式(4.15)所示一阶无限冲激响应低通滤波器,设计一个在 0.3π 处具有 3 dB 截止频率 ω_c 的无限冲激响应低通滤波器。把它与一个具有相同截止频率的一阶无限冲激响应低通滤波器的增益响应做比较。

Q4.16 设计一个中心频率 ω_o 在 0.61π 处、3 dB 带宽为 0.15π 的二阶无限冲激响应带通滤波器。由于式(4.20)是 α 的二次方程,为了产生相同的 3 dB 带宽,参数 α 将有两个数值,得到的传输函数 $H_{BP}(z)$ 也会有两个不同的表达式。使用函数 zplane 可产生两个传输函数的极零点图,从中可选择一个稳定的传输函数。用 MATLAB 计算并画出你所设计的滤波器的增益响应,并验证它确实满足给定的条件。用设计的稳定无限冲激

响应带通滤波器的传输函数的参数 α 和 β,生成一个二阶无限冲激响应带阻滤波器的传输函数 $H_{BS}(z)$。用 MATLAB 证明 $H_{BP}(z)$和 $H_{BS}(z)$都是全通互补和功率互补的。

若 $H(z)$是一个有限冲激响应或无限冲激响应数字滤波器的传输函数,将 L 个延时取代 $H(z)$中的每一个延时,得到的滤波器的传输函数为 $G(z)=H(z^L)$。因此,新滤波器的频率响应是周期为 $2\pi/L$ 的 ω 的周期函数。这种滤波器通常称为梳状滤波器,它会在周期干扰消除中得到应用。

习题

Q4.17 用 MATLAB 计算并画出一个梳状滤波器的幅度响应,该梳状滤波器是在 L 取不同值的情况下,由式(4.40)给出的原型有限冲激响应低通滤波器得到的。证明新滤波器的幅度响应在 $\omega=\omega_k=(2k+1)\pi/L$ 处有 L 个极小值,在 $\omega=\omega_k=2k\pi/L$ 处有 L 个极大值,$k=0,1,\cdots,L-1$。

Q4.18 用 MATLAB 计算并画出一个梳状滤波器的幅度响应,该梳状滤波器是在 L 取不同值的情况下,由式(4.42)在 $M=2$ 时给出的原型有限冲激响应低通滤波器得到的。确定这种梳状滤波器冲激响应的极大值和极小值的位置。

在许多应用中,要求使用线性相位或零相位的数字滤波器。零相位滤波器不可能用一个因果数字滤波器实现。因此图 4.2 中所示的非因果有限冲激响应或无限冲激响应数字滤波器可用于零相位滤波。MATLAB 中的 M 文件 `filtfilt` 可以实现这种零相位滤波。它也可用来减小启动瞬态。由于它在前向和后向上使用相同的滤波器,因此所实现的滤波器的阶数是基本滤波器的两倍,这都需要在实际应用中加以考虑。在第 7 章中,我们将设计并讨论具有精确线性相位的有限冲激响应滤波器。我们已经定义了四种类型的线性相位有限冲激响应滤波器(见 R4.24)。程序 P4.3 用于研究这些滤波器的特性,它先产生这四类滤波器的冲激响应,然后产生极零点图,最终显示出零点位置。

```
%程序 P4.3
%线性相位有限冲激响应滤波器的零点位置
clf;
b=[1 -8.5 30.5 -63];
num1=[b 81 fliplr(b)];
num2=[b 81 81 fliplr(b)];
num3=[b 0 -fliplr(b)];
num4=[b 81 -81 -fliplr(b)];
n1=0:length(num1)-1;
n2=0:length(num2)-1;
subplot(2,2,1); stem(n1,num1);
xlabel('时间序号 n');ylabel('振幅'); grid;
title('1 型有限冲激响应滤波器');
subplot(2,2,2); stem(n2,num2);
xlabel('时间序号 n');ylabel('振幅'); grid;
title('2 型有限冲激响应滤波器');
subplot(2,2,3); stem(n1,num3);
xlabel('时间序号 n');ylabel('振幅'); grid;
title('3 型有限冲激响应滤波器');
subplot(2,2,4); stem(n2,num4);
```

```
xlabel('时间序号 n');ylabel('振幅'); grid;
title('4 型有限冲激响应滤波器');
pause
subplot(2,2,1); zplane(num1,1);
title('1 型有限冲激响应滤波器');
subplot(2,2,2); zplane(num2,1);
title('2 型有限冲激响应滤波器');
subplot(2,2,3); zplane(num3,1);
title('3 型有限冲激响应滤波器');
subplot(2,2,4); zplane(num4,1);
title('4 型有限冲激响应滤波器');
disp('1 型有限冲激响应滤波器的零点是');
disp(roots(num1));
disp('2 型有限冲激响应滤波器的零点是');
disp(roots(num2));
disp('3 型有限冲激响应滤波器的零点是');
disp(roots(num3));
disp('4 型有限冲激响应滤波器的零点是');
disp(roots(num4));
```

习题

Q4.19　运行程序 P4.3,生成每一类线性相位有限冲激响应滤波器的冲激响应。每一个有限冲激响应滤波器的长度是多少?验证冲激响应序列的对称性。接着验证这些滤波器的零点位置。使用 MATLAB 计算并绘出这些滤波器的相位响应,验证它们的线性相位特性。这些滤波器的群延迟是多少?

Q4.20　用 b=[1.5 -3.25 5.25 -4]替换程序 P4.3 中的向量 b,重做习题 Q4.19。

如第 9 章的演示所示,因果系统传输函数的有界实特性是实现系统低通敏感性的关键。为了检测有界实特性,必须先检测传输函数的稳定性,再确定其幅度响应的最大值。注意,任何因果稳定系统的传输函数可通过简单的尺度缩放来使之成为一个有界实函数。

习题

Q4.21　用 MATLAB 确定如下传输函数是否是有界实函数:

$$H_1(z)=\frac{1.5(1+z^{-1})}{1+5z^{-1}+6z^{-2}} \tag{4.43}$$

若它不是一个有界实函数,求一个与 $H_1(z)$有着相同幅度的有界实函数 $H_2(z)$。

Q4.22　用 MATLAB 确定如下传输函数是否是有界实函数:

$$G_1(z)=\frac{1-z^{-1}}{2+z^{-1}+z^{-2}} \tag{4.44}$$

若它不是一个有界实函数,求一个与 $G_1(z)$有着相同幅度的有界实函数 $G_2(z)$。

4.6　稳定性测试

因果无限冲激响应数字滤波器的稳定性是一个重要的设计指标。若因果无限冲激响应滤

波器的传输函数的极点全部在单位圆内,则它就是稳定的。MATLAB 函数 zplane 可用于检测无限冲激响应传输函数的极点位置。然而,若系统有一个或多个极点非常靠近单位圆或者在单位圆上,则通过极零点图检测相应的传输函数的稳定性是不够的。R4.32 中给出了一个更精确的稳定性检测算法。程序 P4.4 使用函数 poly2rc 实现了这种稳定性检测。

```
% 程序 P4.4
% 稳定性检测
clf;
den = input('分母系数 = ');
ki = poly2rc(den);
disp('稳定性测试参数是');
disp(ki);
```

习题

Q4.23 用 MATLAB 产生如下两个因果系统传输函数的极零点图:

$$H_1(z) = \frac{1}{1 - 1.848\,z^{-1} + 0.85\,z^{-2}}$$

$$H_2(z) = \frac{1}{1 - 1.851\,z^{-1} + 0.85\,z^{-2}}$$

研究生成的极零点图,你可以推断它们的稳定性吗?

Q4.24 用程序 P4.4 检测习题 Q4.23 中的两个传输函数的稳定性。这两个传输函数中哪一个是稳定的?

Q4.25 用程序 P4.4 确定下面这个多项式的所有根是否都在单位圆内:

$$D(z) = 1 + 2.5\,z^{-1} + 2.5\,z^{-2} + 1.25\,z^{-3} + 0.3125\,z^{-4} + 0.031\,25\,z^{-5}$$

Q4.26 用程序 P4.4 确定下面这个多项式的所有根是否都在单位圆内:

$$D(z) = 1 + 0.2\,z^{-1} + 0.3\,z^{-2} + 0.4\,z^{-3} + 0.5\,z^{-4} + 0.6\,z^{-5}$$

4.7 基础知识阅读

[1] A. Antoniou. *Digital Filters: Analysis, Design, and Applications.* McGraw-Hill, New York NY, second edition, 1993. Ch. 3.

[2] E. Cunningham. *Digital Filtering: An Introduction.* Houghton-Mifflin, Boston MA, 1992. Ch. 8.

[3] D.J. DeFatta, J.G. Lucas, and W.S. Hodgkiss. *Digital Signal Processing: A System Design Approach.* Wiley, New York NY, 1988. Ch. 9.

[4] L.B. Jackson. *Digital Filters and Signal Processing.* Kluwer, Boston MA, third edition, 1996. Ch. 11.

[5] R. Kuc. *Introduction to Digital Signal Processing.* McGraw-Hill, New York NY, 1988. Ch. 10.

[6] L.C. Ludeman. *Fundamentals of Digital Signal Processing*. Harper & Row, New York NY, 1986. Secs. 1.3.5, 2.3, 2.4.

[7] S.K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, New York NY, 1998. Sec. 8.4 and Ch. 9.

[8] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs NJ, 1989. Secs. 6.7–6.10.

[9] S.J. Orfanidis. *Introduction to Signal Processing*. Prentice-Hall, Englewood Cliffs NJ, 1996. Ch. 2, Sec. 7.6.

[10] B. Porat. *A Course in Digital Signal Procesing*. Wiley, New York NY, 1996. Secs. 2.7, 7.6, 8.1, 8.2, 8.4, 9.1.

[11] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs NJ, third edition, 1996. Secs. 1.4, 7.5–7.7.

[12] R.A. Roberts and C.T. Mullis. *Digital Signal Processing*. Addison-Wesley, Reading MA, 1987. Ch. 9.

第 5 章　连续时间信号的数字处理

5.1　导言

数字信号处理算法经常用来处理连续时间信号。为此，需要将连续时间信号转换成为等效的离散时间信号，对离散信号应用必要的数字信号处理算法，然后将处理后的离散时间信号转换成为等效的连续时间信号。理想情况下，连续时间信号到离散时间形式的转换通过周期抽样实现，并且为了防止混叠，一种模拟反混叠滤波器常常放在抽样之前来带限连续时间信号。离散时间信号转换成连续时间信号需要一个模拟重构滤波器。在这个练习中，我们将分别研究在时域和频域中抽样的效果，以及模拟滤波器的设计。另外，我们还将学会模数和数模转换的基础。

5.2　基础知识回顾

R5.1　设 $g_a(t)$是一个连续时间信号，它在 $t=nT$ 处被均匀抽样，以产生序列 $g[n]$，其中

$$g[n]=g_a(nT), \qquad -\infty<n<\infty \tag{5.1}$$

T 是抽样周期。T 的倒数称为抽样频率 F_T，即 $F_T=1/T$。现在，$g_a(t)$的频域表示由其连续时间傅里叶变换 $G_a(j\Omega)$给出，

$$G_a(j\Omega)=\int_{-\infty}^{\infty}g_a(t)\,e^{-j\Omega t}\,dt \tag{5.2}$$

而 $g[n]$的频域表示由其离散时间傅里叶变换 $G(e^{j\omega})$给出，

$$G(e^{j\omega})=\sum_{n=-\infty}^{\infty}g[n]\,e^{-j\omega n} \tag{5.3}$$

$G_a(j\Omega)$与 $G(e^{j\omega})$之间的关系为

$$\begin{aligned}G(e^{j\omega})&=\frac{1}{T}\sum_{k=-\infty}^{\infty}G_a(j\Omega-jk\Omega_T)|_{\Omega=\omega/T}\\&=\frac{1}{T}\sum_{k=-\infty}^{\infty}G_a(j\frac{\omega}{T}-jk\Omega_T)=\frac{1}{T}\sum_{k=-\infty}^{\infty}G_a(j\frac{\omega}{T}-j\frac{2\pi k}{T})\end{aligned} \tag{5.4}$$

也可表示为

$$G(e^{j\Omega T})=\frac{1}{T}\sum_{k=-\infty}^{\infty}G_a(j\Omega-jk\Omega_T) \tag{5.5}$$

R5.2　抽样定理。设 $g_a(t)$是带限信号，即$|\Omega|>\Omega_m$ 时 $G_a(j\Omega)=0$。则 $g_a(t)$可由其样本 $g_a(nT)\,n=0,1,2,3,\cdots$，惟一确定，若

$$\Omega_T > 2\Omega_m \tag{5.6}$$

其中，

$$\Omega_T = \frac{2\pi}{T} \tag{5.7}$$

给定$\{g[n]\} = \{g_a(nT)\}$，我们可通过产生如下形式的冲激串 $g_p(t)$来正确地恢复 $g_a(t)$：

$$g_p(t) = g_a(t)\,p(t) = \sum_{n=-\infty}^{\infty} g_a(nT)\delta(t - nT) \tag{5.8}$$

然后将 $g_p(t)$穿过增益为 T、截止频率 Ω_c 大于 Ω_m 而小于 $\Omega T - \Omega_m$ 的一个理想低通滤波器 $H_r(j\Omega)$。截止频率具体为

$$\Omega_m < \Omega_c < (\Omega_T - \Omega_m) \tag{5.9}$$

$g_a(t)$的最高频率 Ω_m 通常称为奈奎斯特频率，它决定了从抽样形式中完全恢复 $g_a(t)$所需要的最小抽样频率 $\Omega_T > 2\Omega_m$ 。频率 $2\Omega_m$ 称为奈奎斯特率。

若抽样率高于奈奎斯特率，则称为过抽样。另一方面，若抽样率比奈奎斯特率低，则称为欠抽样。最后，若抽样率恰好等于奈奎斯特率，则称为临界抽样。

R5.3　现在，一个理想模拟低通滤波器的冲激响应 $h_r(t)$可以通过对其频率响应 $H_r(j\Omega)$：

$$H_r(j\Omega) = \begin{cases} T, & |\Omega| \leqslant \Omega_c \\ 0, & |\Omega| > \Omega_c \end{cases} \tag{5.10}$$

进行傅里叶逆变换而简单地得到，即

$$\begin{aligned} h_r(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} H_r(j\Omega)\, e^{j\Omega t}\, dt \\ &= \frac{T}{2\pi}\int_{-\Omega_c}^{\Omega_c} e^{j\Omega t} d\Omega = \frac{\sin(\Omega_c t)}{\Omega_T t/2}, \quad -\infty \leqslant t \leqslant \infty \end{aligned} \tag{5.11}$$

则冲激串 $g_p(t)$为

$$g_p(t) = \sum_{n=-\infty}^{\infty} g[n]\, \delta(t - nT) \tag{5.12}$$

因此，理想低通滤波器的输出 $\hat{g}_a(t)$由 $g_p(t)$和模拟重构滤波器的冲激响应 $h_r(t)$的卷积给出：

$$\hat{g}_a(t) = \sum_{n=-\infty}^{\infty} g[n]\, h_r(t - nT) \tag{5.13}$$

将式(5.11)中的 $h_r(t)$代入式(5.13)中，并假定 $\Omega_c = \Omega_T/2 = \pi/T$，可得

$$\hat{g}_a(t) = \sum_{n=-\infty}^{\infty} g[n]\, \frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T} \tag{5.14}$$

R5.4　滤波器指标通常按照其幅度响应表述。例如，模拟低通滤波器的幅度$|H_a(j\Omega)|$通常如图 5.1 所示。在通带 $0 \leqslant \Omega \leqslant \Omega_p$ 中，我们要求

$$1 - \delta_p \leqslant |H_a(j\Omega)| \leqslant 1 + \delta_p, \quad |\Omega| \leqslant \Omega_p \tag{5.15}$$

或者,换句话说,幅度以误差 $\pm\delta_p$ 逼近1。在阻带 $\Omega_s \leqslant |\Omega| \leqslant \infty$ 中,我们要求

$$|H_a(j\Omega)| \leqslant \delta_s, \quad \Omega_s \leqslant |\Omega| \leqslant \infty \tag{5.16}$$

幅度以误差 δ_s 逼近零。频率 Ω_p 和 Ω_s 分别称为通带边界频率和阻带边界频率。在通带和阻带中的容限 δ_p 和 δ_s 称为波纹。

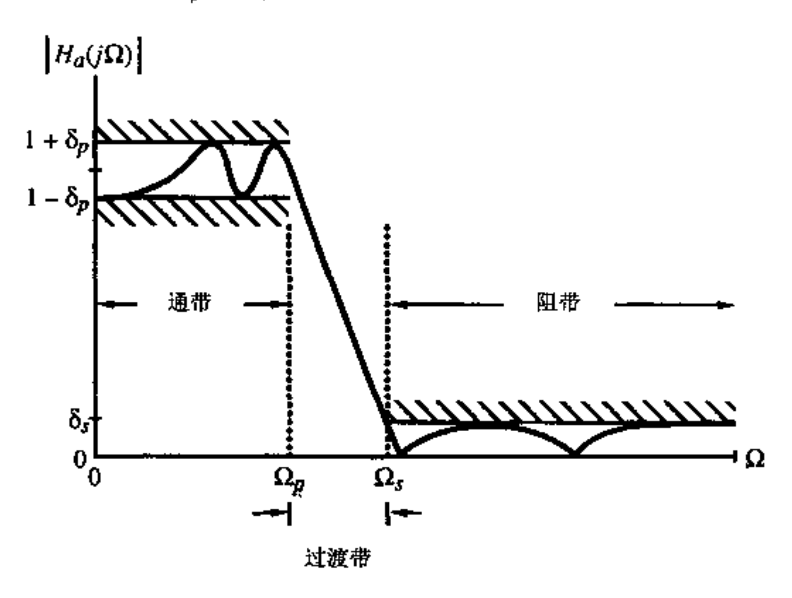


图 5.1 模拟低通滤波器的典型幅度响应指标

R5.5 在大多数应用中,模拟滤波器的指标如图 5.2 所示。此时,在由 $0 \leqslant \Omega \leqslant \Omega_p$ 定义的通带中,幅度响应的最大值及最小值分别是 1 和 $1/\sqrt{1+\epsilon^2}$。峰通带波纹是

$$R_p = 20\log_{10}\sqrt{1+\epsilon^2}\ \text{dB} \tag{5.17}$$

在由 $\Omega_s \leqslant \Omega \leqslant \infty$ 定义的阻带中,最大阻带波纹用 $1/A$ 表示。最小阻带衰减因此为

$$R_s = 20\log_{10} A\ \text{dB} \tag{5.18}$$

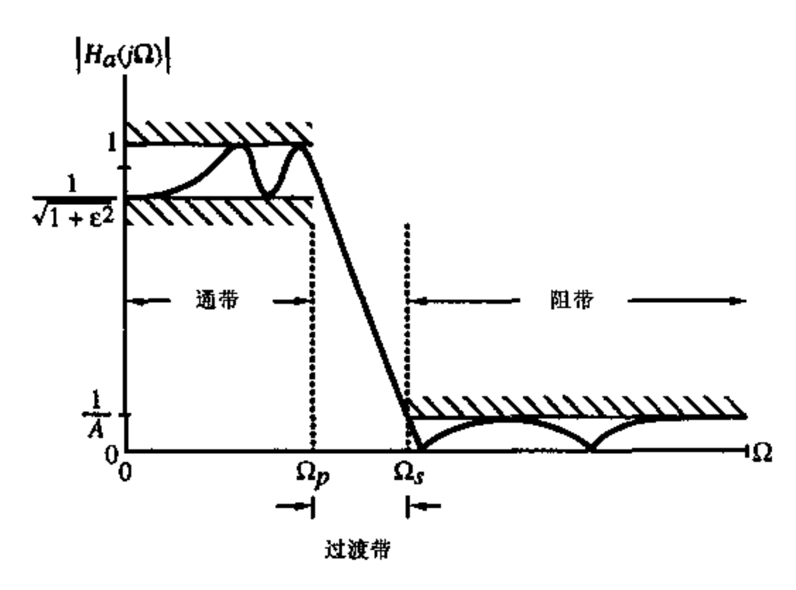


图 5.2 模拟低通滤波器的归一化幅度响应指标

R5.6　巴特沃兹逼近。N 阶模拟低通巴特沃兹滤波器 $H_a(s)$ 的幅度平方响应为

$$|H_a(j\Omega)|^2 = \frac{1}{1+(\frac{\Omega}{\Omega_c})^{2N}} \tag{5.19}$$

若 $|H_a(j\Omega)|^2$ 在 $\Omega=0$ 处的前 $2N-1$ 个导数等于零，则巴特沃兹滤波器在 $\Omega=0$ 处有一个最大的平坦幅度。$\Omega=\Omega_c$ 处的增益 $\mathcal{G}(\Omega)=10\log_{10}|H_a(j\Omega)|^2$ 比 $\Omega=0$ 处的增益小3 dB，因此，Ω_c 称为 3 dB 截止频率。

完全描述巴特沃兹低通滤波器的两个参数是 3 dB 截止频率 Ω_c 和阶数 N。它们由给定的通带边界 Ω_p、阻带边界 Ω_s、单位为 dB 的峰通带波纹 R_p 以及单位为 dB 的最小阻带衰减 R_s 确定。

巴特沃兹低通滤波器的传输函数形如

$$H_a(s) = \frac{K}{\sum_{\ell=0}^{N} a_\ell s^\ell} = \frac{K}{\prod_{\ell=1}^{N}(s-p_\ell)} \tag{5.20}$$

R5.7　切比雪夫 1 型逼近。切比雪夫 1 型低通传输函数 $H_a(s)$ 的幅度响应为

$$|H_a(j\Omega)|^2 = \frac{1}{1+\varepsilon^2 T_N^2(\omega-\Omega_p)} \tag{5.21}$$

其中 $T_N(\Omega)$ 是 N 阶切比雪夫多项式：

$$T_N(\Omega) = \begin{cases} \cos(N\cos^{-1}\Omega), & |\Omega| \leqslant 1 \\ \cosh(N\cosh^{-1}\Omega), & |\Omega| > 1 \end{cases} \tag{5.22}$$

上面的多项式也可由下面给出的递归关系得到：

$$T_r(\Omega) = 2\Omega T_{r-1}(\Omega) - T_{r-2}(\Omega), \quad r \geqslant 2 \tag{5.23}$$

其中 $T_0(\Omega)=1$，$T_1(\Omega)=\Omega$。

切比雪夫 1 型低通滤波器的阶数 N 由给定的通带边界 Ω_p、阻带边界 Ω_s、单位为 dB 的峰通带波 R_p 以及单位为 dB 的最小阻带衰减 R_s 确定。同样可得传输函数 $H_a(s)$ 如式(5.20)所示。

R5.8　切比雪夫 2 型逼近。平方幅度响应表达式为

$$|H_a(j\Omega)|^2 = \frac{1}{1+\varepsilon^2\left[\frac{T_n(\Omega_s/\Omega_p)}{T_n(\Omega_s/\Omega)}\right]^2} \tag{5.24}$$

由于切比雪夫 2 型低通滤波器的传输函数既有极点也有零点，它不再是一个全极点函数。其形式为

$$H_a(s) = K\frac{\sum_{\ell=0}^{N} b_\ell s^\ell}{\sum_{\ell=0}^{N} a_\ell s^\ell} = K\frac{\prod_{\ell=1}^{N}(s-z_\ell)}{\prod_{\ell=1}^{N}(s-p_\ell)} \tag{5.25}$$

零点 z_ℓ 此时在 $j\Omega$ 轴上。切比雪夫 2 型低通滤波器的阶数 N 由给定的通带边界 Ω_p、阻带边界 Ω_s、单位为 dB 的峰通带波纹 R_p 以及单位为 dB 的最小阻带衰减 R_s 确定。

R5.9　椭圆逼近。椭圆低通滤波器的平方幅度响应为

$$|H_a(j\Omega)|^2 = \frac{1}{1+\varepsilon^2 R_N^2(\Omega/\Omega_p)} \tag{5.26}$$

其中 $R_N(\Omega)$是满足性质 $R_N(1/\Omega) = 1/R_N(\Omega)$的阶数为 N 的有理数函数,其分子的根在区间 $0<\Omega<1$ 内,而分母的根位于区间 $1<\Omega<\infty$内。椭圆低通滤波器的阶数 N 由给定的通带边界 Ω_p、阻带边界 Ω_s、单位为 dB 的峰通带波纹 R_p 以及单位为 dB 的最小阻带衰减 R_s 确定。

R5.10 滤波器类型的比较。接下来通过检测同样阶数的归一化巴特沃兹、切比雪夫以及椭圆模拟低通滤波器的频率响应,来比较上述类型近似的性能。假定切比雪夫1型和等波滤波器的通带波纹是一样的,同时假定切比雪夫2型和等波滤波器的最小阻带衰减是一样的。用来比较的滤波器指标如下:滤波器阶数为6、通带边界在 $\Omega=1$ 处、最大通带偏离为1 dB 、最小阻带衰减为40 dB。图5.3画出了用 MATLAB 计算的频率响应。

由图5.3可看出,巴特沃兹滤波器有最宽的过渡带,它具有单调递减的增益响应。两种类型的切比雪夫滤波器有相等宽度的过渡带,该宽度小于巴特沃兹滤波器但大于椭圆滤波器。切比雪夫1型滤波器在过渡带中提供一个比切比雪夫2型滤波器稍微快一些的下降。在通带中,切比雪夫2型滤波器的幅度响应与巴特沃兹滤波器的几乎一样。椭圆滤波器有最窄的过渡带并具有等波纹通带和等波纹阻带响应。

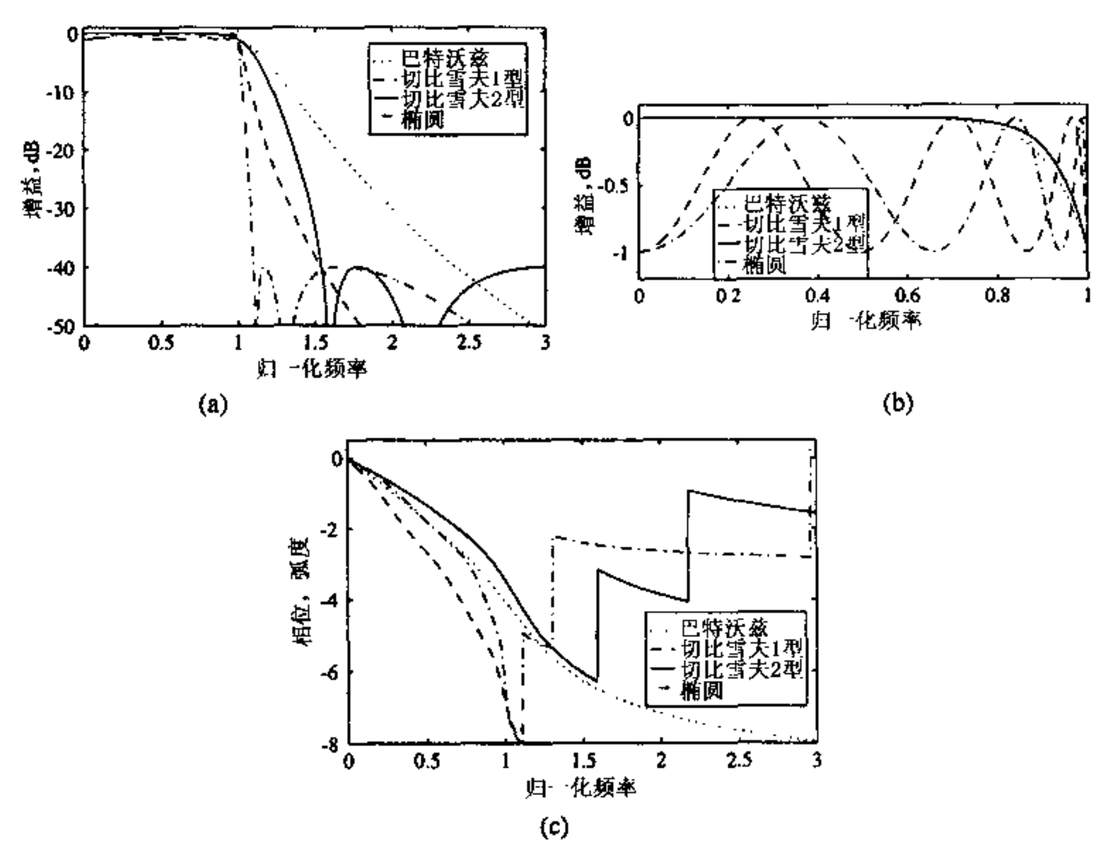


图5.3 四种类型模拟低通滤波器的频率响应的比较:(a)增益响应,(b)通带细节,(c)相位响应

巴特沃兹和切比雪夫滤波器在通带大约四分之三的地方都有几乎线性的相位响应,而椭圆滤波器在通带大约一半的地方有几乎线性的相位响应。

R5.11 二进制数字表示法。在二进制表示中,用称为比特的符号 0 和 1 表示一个数字。二进制点将整数部分和小数部分分开。一位的附加符号位在整数部分的左边,以表示数字的符号。对于正数,符号位是 0,而对于负数,符号位是 1。三个常用的二进制数字表示形式是原码、反码和补码形式。

含有 I 个整数位、F 个小数位和一个符号位的二进制数字形如 $sa_{I-1}a_{I-2}\cdots a_1a_0\Delta a_{-1}a_{-2}\cdots a_{-F}$,其中 Δ 表示二进制点,a_k 表示每一个位,符号位 s 是 0 或 1。位 a_{I-1}称为最高位,缩写为 MSB,而位 a_{-F}称为最低位,缩写为 LSB。在定点二进制表示中,对于所有数字,二进制点的位置总是在同一地方。在数字信号处理应用中,定点数总是以小数表示。

若 $s=0$,原码形式的二进制小数 $s\Delta a_{-1}a_{-2}\cdots a_{-F}$的十进制等效是正数,其幅度为 $\sum_{k=1}^{F}a_{-k}2^{-k}$,而若 $s=1$,则十进制等效为负数,其幅度为$\sum_{k=1}^{F}a_{-k}2^{-k}$。

在反码形式中,正或负小数 $s\Delta a_{-1}a_{-2}\cdots a_{-F}$的十进制等效为 $-s(1-2^{-F})+\sum_{k=1}^{F}a_{-k}2^{-k}$。在这种形式中,正小数的表示与原码形式的表示相同,但其相反数通过对原码形式中的正小数的二进制表示的每一位取补来表示(即用 1 代替每个 0,反之亦然)。

在补码形式中,正或负小数 $s\Delta a_{-1}a_{-2}\cdots a_{-F}$的十进制等效为 $-s+\sum_{k=1}^{F}a_{-k}2^{-k}$。在这种形式中,正小数的表示与原码形式的表示相同,但其相反数通过对原码形式中的正小数的二进制表示的每一位取补,然后在最低有效位(第 F 位)上加一个 1 得到。

R5.12 一种实际的数模转换器首先从数字输入信号中生成一个模拟周期性脉冲串,然后通过一个零阶保持电路将它转换成一个类阶梯的模拟波形 $y_z(t)$。零阶保持电路的连续时间傅里叶变换为

$$H_z(j\Omega)=e^{-j\frac{\Omega T}{2}}\left[\frac{sin(\Omega T/2)}{\Omega/2}\right] \tag{5.27}$$

其中 T 是数字信号的抽样周期。零阶保持电路因此具有这样一个幅度响应:零点在 $\Omega=1/T$ 的整数倍处,且具有低通特性。在模数与数模转换器之间,会导致离散时间系统的幅度响应的振幅失真,这称为*偏差*。偏差可通过用下面给出的频率响应设计模拟重构滤波器来补偿:

$$\hat{H}_r(j\Omega)=\frac{H_r(j\Omega)}{H_z(j\Omega)} \tag{5.28}$$

其中 $H_r(j\Omega)$是理想模拟重构低通滤波器的频率响应。另外,偏差可通过在数模转换之前加入一个幅度响应是零阶保持电路的逆的数字补偿滤波器来补偿。

5.3 用到的 MATLAB 命令

这个练习用到的 MATLAB 命令如下:

通用命令

```
length    size
```

```
title('连续时间信号 x_{a}(t)');
axis([0 1 -1.2 1.2])
subplot(2,1,2);
T = 0.1;
n = 0:T:1;
xs = cos(2*pi*f*n);
k = 0:length(n)-1;
stem(k,xs); grid
xlabel('时间序号 n');ylabel('振幅');
title('离散时间信号 x[n]');
axis([0 (length(n)-1) -1.2 1.2])
```

习题

Q5.1 运行程序 P5.1,产生连续时间信号及其抽样形式,并显示它们。

Q5.2 正弦信号的频率是多少赫兹?抽样周期是多少秒?

Q5.3 解释两个 axis 命令的效果。

Q5.4 以比在程序 P5.1 中列出的抽样周期低的两个抽样周期和高的两个抽样周期的四个其他值,运行程序 P5.1。评论你的结果。

Q5.5 通过将正弦信号的频率分别变为 3 Hz 和 7 Hz,重做习题 P5.1。相应的等效离散时间信号与习题 Q5.1 中产生的离散时间信号之间有差别吗?若没有,为什么没有?

例 5.2　时域中的混叠效果

在本例中,我们将产生程序 P5.1 所生成的离散时间信号 $x[n]$的连续时间等效 $y_a(t)$,从而研究正弦信号 $x_a(t)$的频率与抽样周期之间的关系。为了从 $x[n]$中产生重构的信号 $y_a(t)$,我们将 $x[n]$通过一个理想低通滤波器,该滤波器可按照式(5.11)来实现(见 R5.3)。若式(5.11)在间隔很密的 t 值处计算,$y_a(t)$的曲线将类似一个连续时间信号。为了在 MATLAB 上实现这个方程,式(5.11)中的和需要用一个有限和代替,因此可以产生所期望的重构连续时间信号 $y_a(t)$的一个逼近。

```
% 程序 P5.2
% 时域中混叠效果的说明
% 获得了 Mathwork 公司的允许
% 本程序由[Kra94]改编
clf;
T = 0.1;f = 13;
n = (0:T:1)';
xs = cos(2*pi*f*n);
t = linspace(-0.5,1.5,500)';
ya = sinc((1/T)*t(:,ones(size(n))) - (1/T)*n(:,ones(size(t)))')*xs;
plot(n,xs,'o',t,ya);grid;
xlabel('时间,msec');ylabel('振幅');
title('重构的连续时间信号 y_{a}(t)');
axis([0 1 -1.2 1.2])
```

习题

Q5.6 运行程序 P5.2,产生离散时间信号 $x[n]$及其连续时间等效 $y_a(t)$,并显示它们。

Q5.7 在程序 P5.2 中,t 的范围和时间增量的值是什么?在图中,t 的范围是什么?改变 t 的范围,显示上述程序所计算的全范围 $y_a(t)$并再次运行程序 P5.2。评论这种改变后产生的曲线。

Q5.8 恢复原始显示范围并通过分别改变正弦信号的频率到 3 Hz 和 7 Hz,重复程序 P5.2。相应的等效离散时间信号与习题 Q5.6 中产生的离散时间信号有差别吗?若没有,为什么没有?

5.5 频域中抽样的效果

例 5.3 频域中混叠的效果

在本例中,我们接着研究任意带限连续时间信号的连续时间傅里叶变换(CTFT)与离散时间信号的离散时间傅里叶变换(DTFT)之间的关系。为了将连续时间信号 $x_a(t)$转换成为等效的离散时间信号 $x[n]$,前者必须在频域中带限(见 R5.2)。为了说明在频域中的抽样效果,我们选择一个连续时间傅里叶变换近似带限的指数衰减连续时间信号。

```
%程序 P5.3
%频域中混叠效果
%的说明
clf;
t=0:0.005:10;
xa=2*t.*exp(-t);
subplot(2,2,1)
plot(t,xa);grid
xlabel('时间,msec');ylabel('振幅');
title('连续时间信号 x_{a}(t)');
subplot(2,2,2)
wa=0:10/511:10;
ha=freqs(2,[1 2 1],wa);
plot(wa/(2*pi),abs(ha));grid;
xlabel('频率,kHz');ylabel('振幅');
title('|X_{a}(j\Omega)|');
axis([0 5/pi 0 2]);
subplot(2,2,3)
T=1;
n=0:T:10;
xs=2*n.*exp(-n);
k=0:length(n)-1;
stem(k,xs);grid;
xlabel('时间序号 n');ylabel('振幅');
title('离散时间信号 x[n]');
subplot(2,2,4)
wd=0:pi/225:pi;
hd=freqs(xs,1,wd);
plot(wd/(T*pi),T*abs(hd));grid;
xlabel('频率,kHz');ylabel('振幅');
title('|X(e^{j\omega})|');
axis([0 1/T 0 2]);
```

习题

Q5.9　在程序 P5.3 中,连续时间函数 $x_a(t)$是什么? $x_a(t)$的连续时间傅里叶变换是如何计算的?

Q5.10　运行程序 P5.3,产生并显示离散时间信号及其连续时间等效,以及它们各自的傅里叶变换。有任何明显的混叠影响吗?

Q5.11　将抽样周期增加到 1.5,重复程序 P5.3。有任何明显的混叠影响吗?

Q5.12　对于 $x_a(t)=e^{-\pi t^2}$的情况,修改程序 P5.3 并重做习题 Q5.10 和习题 Q5.11。

5.6　模拟低通滤波器

在连续时间信号的数字处理中,模拟低通滤波器被用做反混叠滤波器和反镜像滤波器。在这一节中,我们将学会在 R5.6 到 R5.9 中总结的四种类型模拟低通滤波器的设计。

例 5.4　模拟低通滤波器的设计

在任何滤波器的设计中,第一步是确定滤波器阶数 N 及适当的截止频率 Ω_c。对于巴特沃兹滤波器,可使用 MATLAB 命令 buttord 来确定这些参数;对于切比雪夫 1 型滤波器,可使用命令 cheb1ord 来确定这些参数;对于切比雪夫 2 型滤波器,可使用命令 cheb2ord 来确定这些参数;对于椭圆滤波器,可使用命令 ellipord 来确定这些参数。Ω_c 对于巴特沃兹滤波器是 3 dB 截止频率,对于切比雪夫 1 型滤波器是通带边界,对切比雪夫 2 型滤波器是阻带边界,而对于椭圆滤波器是通带边界。对于巴特沃兹滤波器,设计滤波器的 MATLAB 命令是 butter;对于切比雪夫 1 型滤波器,设计滤波器的 MATLAB 命令是 cheby1;对于切比雪夫 2 型滤波器,设计滤波器的命令是 cheby2;而对于椭圆滤波器,设计滤波器的命令是 ellip。

程序 P5.4 可用于巴特沃兹低通滤波器的设计。

```
% 程序 P5.4
% 模拟低通滤波器的设计
clf;
Fp = 3500;Fs = 4500;
Wp = 2 * pi * Fp; Ws = 2 * pi * Fs;
[N,Wn] = buttord(Wp,Ws,0.5,30,'s');
[b,a] = butter(N,Wn,'s');
wa = 0:(3 * Ws)/511:3 * Ws;
h = freqs(b,a,wa);
plot(wa/(2 * pi),20 * log10(abs(h)));grid
xlabel('频率,Hz');ylabel('增益,dB');
title('增益响应');
axis([0 3 * Fs -60 5]);
```

习题

Q5.13　在程序 P5.4 中,通带波纹 R_p 和最小阻带衰减 R_s 是多少 dB? 通带及阻带边界频率是多少 Hz?

Q5.14　运行程序 P5.4 并显示增益响应。所设计的滤波器满足给定的指标吗? 所设计的滤波器的阶数 N 和单位为 Hz 的 3 dB 截止频率是多少?

Q5.15 用 cheb1ord 和 cheby1 修改程序 P5.4,以设计与程序 P5.4 有着相同指标的一个切比雪夫 1 型低通滤波器。运行修改的程序并显示增益响应。所设计的滤波器满足给定的指标吗?所设计的滤波器的阶数 N 和单位为 Hz 的通带边界频率是多少?

Q5.16 用 cheb2ord 和 cheby2 修改程序 P5.4,以设计与程序 P5.4 有着相同指标的一个切比雪夫 2 型低通滤波器。运行修改的程序并显示增益响应。所设计的滤波器满足给出的指标吗?所设计的滤波器的阶数 N 和单位为 Hz 的阻带边界频率是多少?

Q5.17 用 ellipord 和 ellip 修改程序 P5.4,以设计与程序 P5.4 有着相同指标的一个椭圆低通滤波器。运行修改的程序并显示增益响应。所设计的滤波器满足给出的规格吗?所设计的滤波器的阶数 N 和单位为 Hz 的通带边界频率是多少?

5.7 模数转换和数模转换

在这一节中,我们将学习模数转换和数模转换,以及十进制数的二进制表示基础。

例 5.5 一个十进制数的二进制等效

程序 P5.5 可用来生成十进制小数的原码形式的二进制等效。

```
% 程序 P5.5
% 确定一个十进制数的原码形式的
% 二进制等效
d = input('输入十进制小数 = ');
b = input('输入所需的字长 = ');
d1 = abs(d);
beq = [zeros(1,b)];
for k = 1:b
    int = fix(2 * d1);
    beq(k) = int;
    d1 = 2 * d1 - int;
end
if sign(d) = = - 1;
    bin = [1 beq];
else
    bin = [0 beq];
end
disp('二进制等效是');
disp(bin)
```

习题

Q5.18 在程序 P5.5 中,运算符 == 的功能是什么?

Q5.19 用程序 P5.5 对于字长值 6 和 8,生成如下十进制小数的原码形式的二进制等效:(a) 0.80165,(b) -0.80165,(c)0.64333,(d) -0.9125。通过手算验证结果。

例 5.6 一个二进制数的十进制等效

程序 P5.6 实现逆过程并产生二进制小数的原码形式的十进制等效。

```
%程序 P5.6
%确定一个二进制小数的
%原码形式的十进制等效
bin = input('输入二进制小数 = ');
b = length(bin) - 1; d = 0;
for k = 1:b
    d = d + bin(k+1) * 2^( - k);
end
if sign(bin(1)) = = 0;
    dec = d;
else
    dec = - d;
end
disp('十进制等效是');
disp(dec)
```

习题

Q5.20 使用程序 P5.6 确定习题 Q5.19 中生成的二进制小数的十进制等效。你的答案和原十进制小数有多接近?

例 5.7　二进制数表示方案

程序 P5.7 可用来确定一个二进制数的原码形式的反码,而程序 P5.8 可用来确定一个负二进制小数的反码形式的补码表示。

```
%程序 P5.7
%确定一个二进制数
%的原码形式的反码等效
bin = input('输入二进制数 = ');
if sign(bin(1)) == 0;
    onescomp = bin;
else
    bin(1) = 0;onescomp = ~bin;
end
disp('反码等效是');
disp(onescomp);

%程序 P5.8
%一个负二进制小数的
%反码形式的补码等效
b = input('输入二进制小数 = ');
F = length(b);
twoscomp = ones(1,F);
c = 1;
for k = F: - 1:2
    if b(k) & c = = 1;
       twoscomp(k) = 0; c = 1;
       else
       twoscomp(k) = b(k) | c; c = 0;
```

```
    end
end
disp('补码等效是 = ');
disp(twoscomp)
```

习题

Q5.21 在程序 P5.7 中,运算符~的目的是什么?

Q5.22 使用程序 P5.7 确定并验证习题 Q5.19 中展开的二进制数的反码表示。

Q5.23 在程序 P5.8 中,运算符|和 & 的目的是什么?

Q5.24 使用程序 P5.8 确定并验证在习题 Q5.19 中生成的二进制数的补码表示。

例 5.8 数模转换器偏差补偿

在这一节中,我们将通过在数模转换器之前嵌入的一个数字滤波器来研究偏差补偿。两个非常简单的低阶下降补偿数字滤波器可用下面的传输函数描述[Jac96]:

$$H_{FIR}(z) = \frac{1}{16}(-1 + 18\,z^{-1} - z^{-2}) \tag{5.29}$$

$$H_{IIR}(z) = \frac{9}{8 + z^{-1}} \tag{5.30}$$

习题

Q5.25 编写一个 MATLAB 程序,确定并在同一幅图上画出未经补偿的及经过偏差补偿的数模转换器的幅度响应。使用式(5.29)和式(5.30)给出的两个有限冲激响应和无限冲激响应偏差补偿滤波器。运行程序并评论你的结果。

5.8 基础知识阅读

[1] A. Antoniou. *Digital Filters: Analysis, Design, and Applications*. McGraw-Hill, New York NY, second edition,1993. Chs. 5, 6.

[2] E. Cunningham. *Digital Filtering: An Introduction*. Houghton-Mifflin, Boston MA, 1992. Secs. 2.4, 3.2, 3.3.

[3] D.J. DeFatta, J.G. Lucas, and W.S. Hodgkiss. *Digital Signal Processing: A System Design Approach*. Wiley, New York NY, 1988. Secs. 2.5, 4.2.

[4] L.B. Jackson. *Digital Filters and Signal Processing*. Kluwer, Boston MA, third edition, 1996. Secs. 6.3, 8.1.

[5] R. Kuc. *Introduction to Digital Signal Processing*, McGraw-Hill, New York, NY, 1988, Secs. 3.10, 3.11.

[6] L.C. Ludeman, *Fundamentals of Digital Signal Processing*. Harper & Row, New York NY, 1986. Secs. 1.5, 1.6 and Ch. 3.

[7] S.K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, New York NY, 1998. Ch. 5 and Sec. 8.4.

[8] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs NJ, 1989. Secs. 3.1–3.3, Appendix B.

[9] S.J. Orfanidis. *Introduction to Signal Processing*. Prentice-Hall, Englewood Cliffs NJ, 1996. Ch. 1.

[10] B. Porat. *A Course in Digital Signal Procesing*. Wiley, New York NY, 1996. Secs. 3.1–3.5, 38. 10.1–10.4.

[11] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs NJ, third edition, 1996. Secs. 1.4.1, 1.4.2, 8.3.5, 9.1.

[12] R.A. Roberts and C.T. Mullis. *Digital Signal Processing*. Addison-Wesley, Reading MA, 1987. Secs. 4.6, 4.7.

R6.5 有限冲激响应滤波器的一种直接型实现,可由式(6.2)生成,$M=5$ 的情况如图 6.2(a)所示。其转置,如图 6.2(b)所示,是第二个直接型结构。通常,一个长度为 M 的有限冲激响应滤波器由 M 个系数描述,并且需要 M 个乘法器和$(M-1)$个双输入加法器来实现。

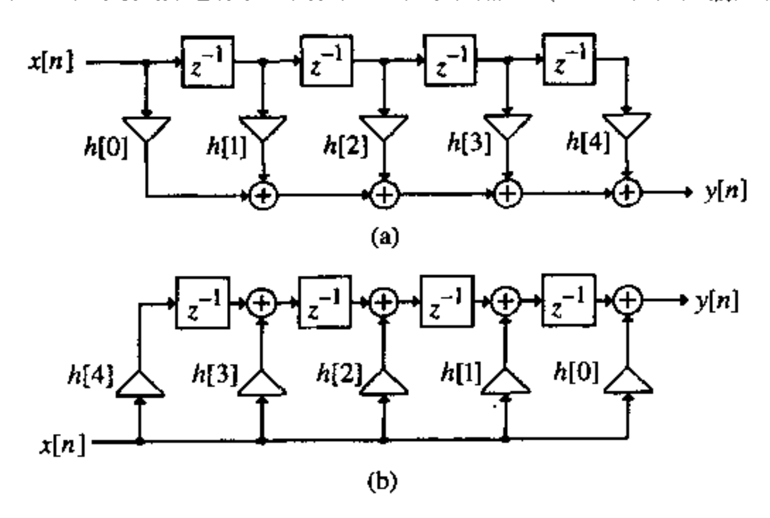


图 6.2　直接型有限冲激响应滤波器结构

R6.6 一个较高阶的有限冲激响应传输函数也可以用有限冲激响应部分的级联实现,其中每个部分用一个一阶或一个二阶传输函数描述。为此,式(6.1)给出的有限冲激响应传输函数 $H(z)$以因式形式表示为

$$H(z)=h[0]\prod_{k}(1+\beta_{1k}z^{-1}+\beta_{2k}z^{-2}) \tag{6.3}$$

其中对一阶因式 $\beta_{2k}=0$。对于三个二阶因式级联,图 6.3 中显示了式(6.3)的一个实现。当然,图 6.3 中的每一个二阶阶段,也可以用转置的直接型实现。通常,对于长度为 M 的有限冲激响应传输函数级联型的实现,也需要$(M-1)$个双输入加法器和 M 个乘法器。

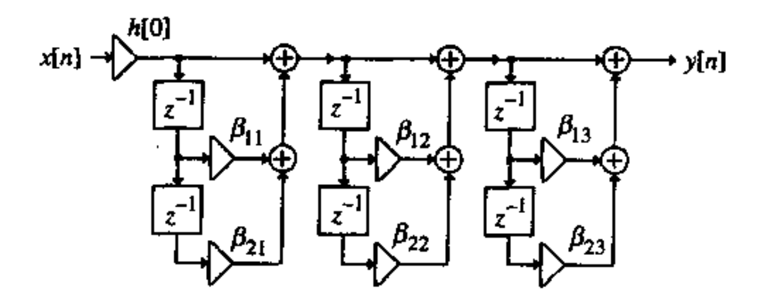


图 6.3　一个长度为 7 的有限冲激响应滤波器的级联型有限冲激响应结构

R6.7 长度为 M 的线性相位有限冲激响应滤波器由对称的冲激响应 $h[n]=h[M-1-n]$或反对称的冲激响应 $h[n]=-h[M-1-n]$描述。利用线性相位有限冲激响应滤波器的对称(或反对称)性质,可以将传输函数的直接型实现所需的乘法器总量减少一半。例如,图 6.4(a)显示了一个具有对称冲激响应的、长度为 7 的 1 型有限冲激响应传输函数的实现,而图 6.4(b)显示了一个具有反对称冲激响应的、长度为 8 的 2 型有限冲激响应传输函数的实现。

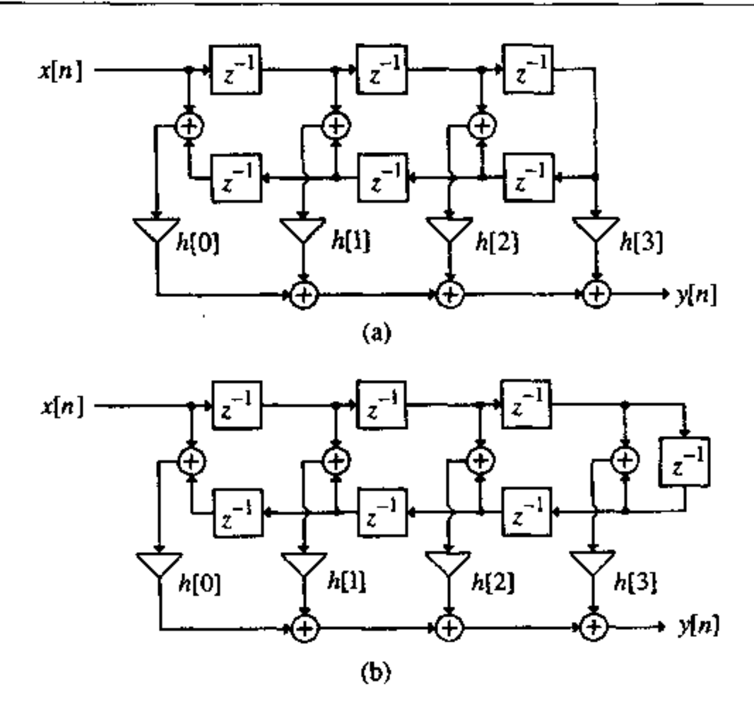


图 6.4　线性相位有限冲激响应结构:(a)1 型;(b)2 型

R6.8　N 阶因果无限冲激响应滤波器由传输函数 $H(z)$:

$$H(z)=\frac{\sum_{k=0}^{N}p_k\,z^{-k}}{1+\sum_{k=1}^{N}d_k\,z^{-k}} \tag{6.4}$$

描述,它是次数为 N 的 z^{-1} 的多项式的比。在时域中,上述无限冲激响应滤波器的输入输出关系为

$$y[n]=\sum_{k=0}^{N}p_k\,x[n-k]-\sum_{k=1}^{N}d_k\,y[n-k] \tag{6.5}$$

其中 $y[n]$ 和 $x[n]$ 分别是输出和输入序列。

R6.9　通过定义一个中间信号变量 $w[n]$:

$$w[n]=\sum_{k=0}^{N}p_k\,x[n-k] \tag{6.6}$$

式(6.5)给出的差分方程可另写为

$$y[n]=w[n]-\sum_{k=1}^{N}d_k\,y[n-k] \tag{6.7}$$

基于式(6.6)和式(6.7)的无限冲激响应的实现称为直接 I 型结构,图 6.5(a)中显示了 $N=3$ 的情况。其转置形式在图 6.5(b)中显示。在直接 I 型实现中,所需要的延时总数是 $2N$。通过简单的框图控制,可将延时总数减少到 N,从而生成直接 II 型结构,$N=3$ 的情况如图 6.6 所示。

一个 N 阶无限冲激响应数字滤波器传输函数由 $2N+1$ 个惟一的系数描述,且通常需要 $2N+1$ 个乘法器和 $2N$ 个双输入加法器来实现。

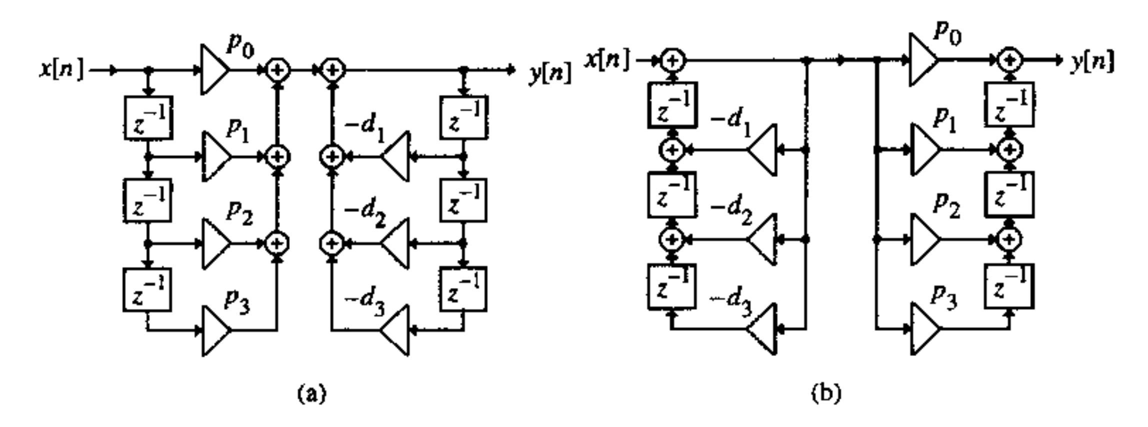


图 6.5　(a)直接Ⅰ型结构,(b)转置的直接Ⅰ型结构

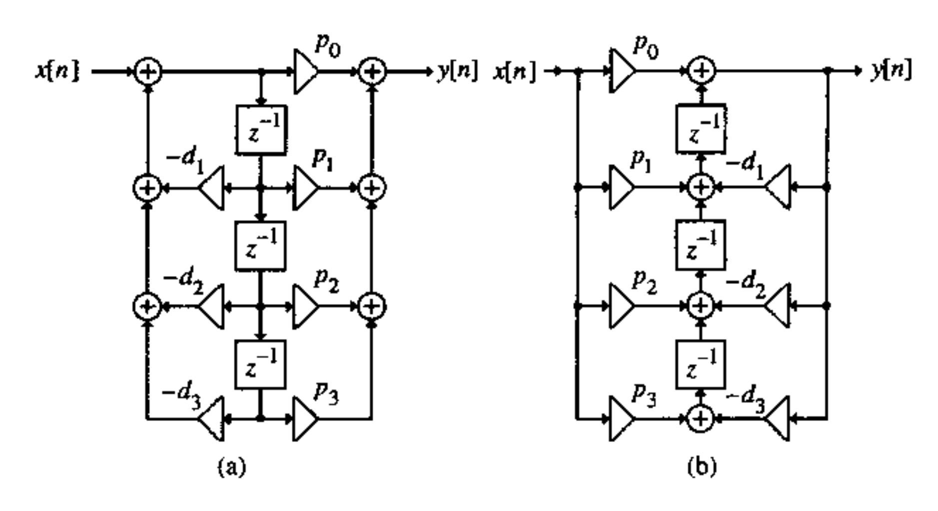


图 6.6　(a)直接Ⅱ型结构,(b)转置的直接Ⅱ型结构

R6.10　通过将传输函数 $H(z)$的分子和分母多项式表示为较低次数多项式的乘积,数字滤波器常常以低阶滤波器部分的级联实现。通常,多项式分解成为一阶和二阶多项式的乘积。在这种情况下,$H(z)$表示为

$$H(z) = p_0 \prod_k \left(\frac{1 + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right) \tag{6.8}$$

在上式中,对于一阶因式 $\alpha_{2k} = \beta_{2k} = 0$。三阶传输函数

$$H(z) = p_0 \left(\frac{1 + \beta_{11} z^{-1}}{1 + \alpha_{11} z^{-1}} \right) \left(\frac{1 + \beta_{12} z^{-1} + \beta_{22} z^{-2}}{1 + \alpha_{12} z^{-1} + \alpha_{22} z^{-2}} \right) \tag{6.9}$$

的一种可能实现如图 6.7 所示。

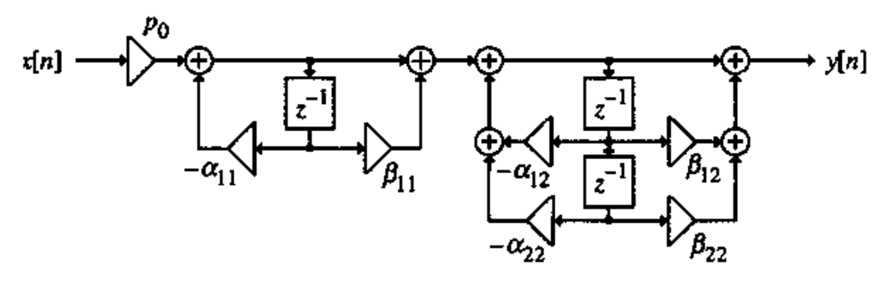


图 6.7　一个三阶无限冲激响应传输函数的级联实现

R6.11 无限冲激响应传输函数可通过使用传输函数的部分分式展开以一种并联形式实现。用 z^{-1} 的传输函数的部分分式展开式得到并联 I 型。因此,假定为单极点,$H(z)$可以表示为

$$H(z)=\gamma_0+\sum_k\left(\frac{\gamma_{0k}+\gamma_{1k}z^{-1}}{1+\alpha_{1k}z^{-1}+\alpha_{2k}z^{-2}}\right) \tag{6.10}$$

在上式中,对于一个实数极点 $\alpha_{2k}=\gamma_{1k}=0$。

传输函数 $H(z)$的一个直接部分分式展开表示为 z 的多项式的比,得到并联结构的第二个基本形式,即并联 II 型[Mit77a]。假定为单极点,$H(z)$可以表示为

$$H(z)=\delta_0+\sum_k\left(\frac{\delta_{1k}z^{-1}+\delta_{2k}z^{-2}}{1+\alpha_{1k}z^{-1}+\alpha_{2k}z^{-2}}\right) \tag{6.11}$$

此时,对于一个实数极点 $\alpha_{2k}k=\beta_{2k}=0$。

一个三阶无限冲激响应传输函数的两个基本并联实现如图 6.8 所示。

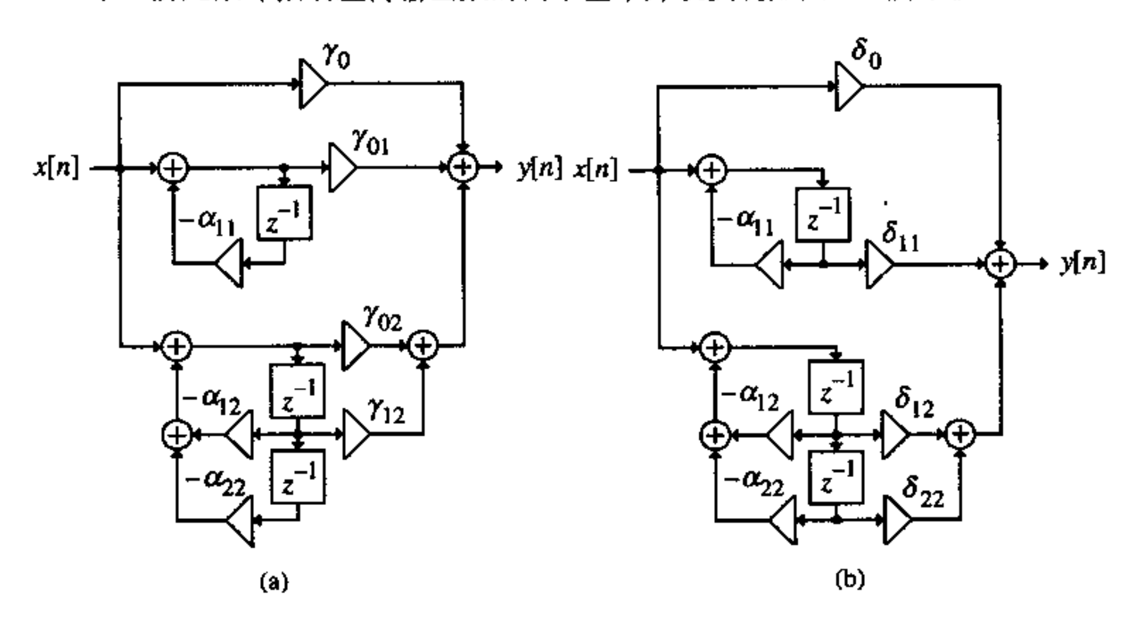


图 6.8 一个三阶无限冲激响应传输函数的并联实现:(a)并联 I 型,(b)并联 II 型

R6.12 M 阶实系数全通传输函数

$$A_M(z)=\frac{d_M+d_{M-1}z^{-1}+\ldots+d_1z^{-(M-1)}+z^{-M}}{1+d_1z^{-1}+\ldots+d_{M-1}z^{-(M-1)}+d_Mz^{-M}} \tag{6.12}$$

由 M 个惟一的系数描述,并可用仅仅 M 个乘法器实现。在一种方法中,$A_M(z)$以二阶和一阶全通部分的级联形式实现。在第二种方法中,$A_M(z)$作为被 $M-1$ 阶的全通传输函数 $A_{M-1}(z)$限制的一个一阶格型二端口网络实现。通过重复这个过程,得到以级联格型结构形式出现的 $A_M(z)$的一个实现。

R6.13 一阶全通传输函数

$$A_1(z)=\frac{d_1+z^{-1}}{1+d_1z^{-1}} \tag{6.13}$$

的两个一次乘法器实现，称为1型全通结构，如图6.9所示[Mit74a]。这些结构的转置产生两个另外的1型全通结构。

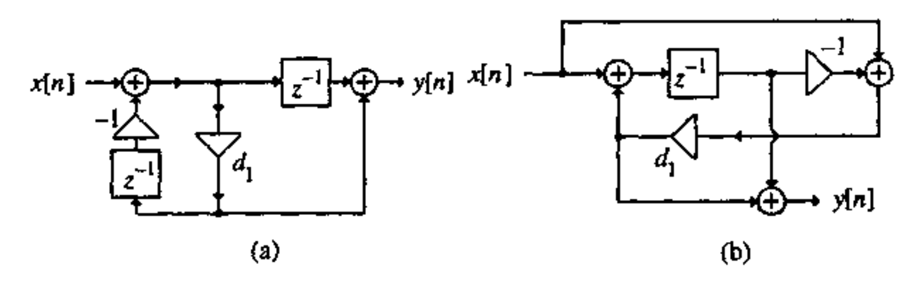


图6.9　(a)1A型全通结构，(b)1B型全通结构

R6.14　形如

$$A_2(z) = \frac{d_1 d_2 + d_1 z^{-1} + z^{-2}}{1 + d_1 z^{-1} + d_1 d_2 z^{-2}} \tag{6.14}$$

的二阶全通传输函数的二次乘法器实现，称为2型全通结构，如图6.10所示[Mit74a]。其他的2型全通结构可通过对这些结构转置得到。

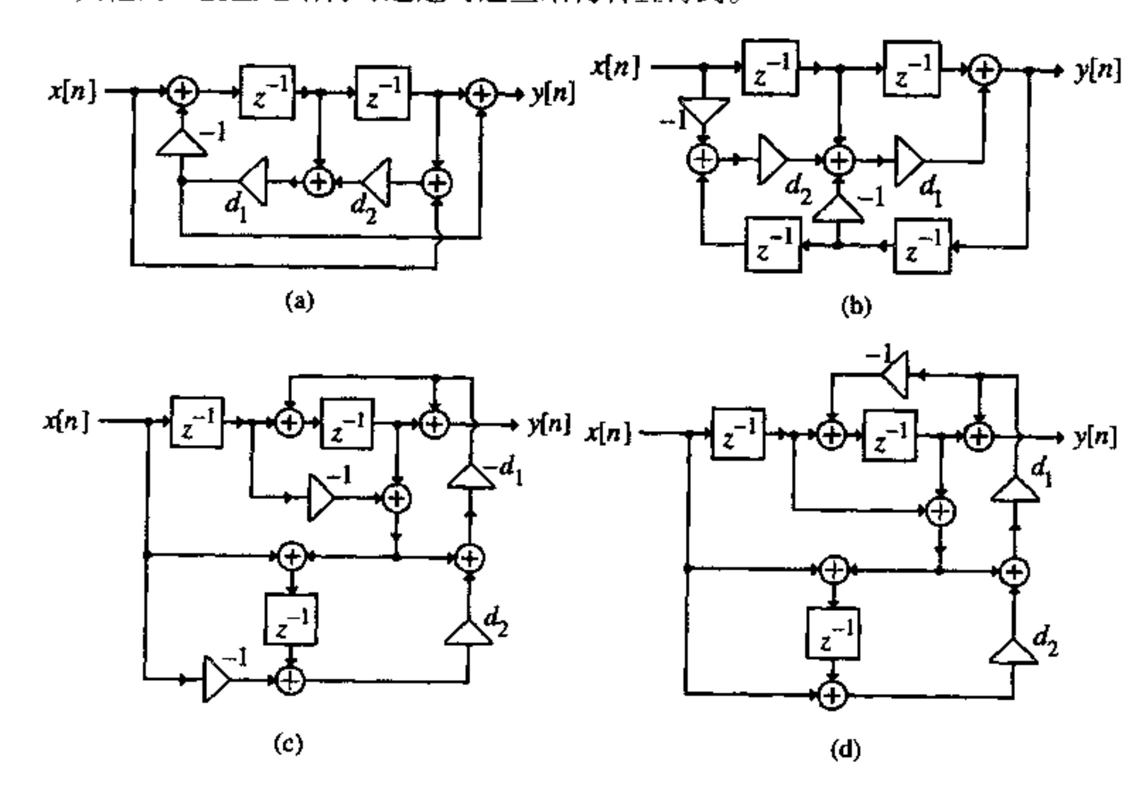


图6.10　(a)2A型全通结构，(b)2D型全通结构，(c)2B型全通结构，(d)2C型全通结构

R6.15　形如

$$A_2(z) = \frac{d_2 + d_1 z^{-1} + z^{-2}}{1 + d_1 z^{-1} + d_2 z^{-2}} \tag{6.15}$$

的二阶全通传输函数的二次乘法器实现，称为3型全通结构，如图6.11所示[Mit74a]。其他的3型全通结构可通过对这些结构转置得到。

R6.16 图6.9至图6.11所示的全通结构的传输函数,对于乘法器系数的任何值保持全通,只要它们是稳定的,就称为是结构上无损有界实的(LBR)。

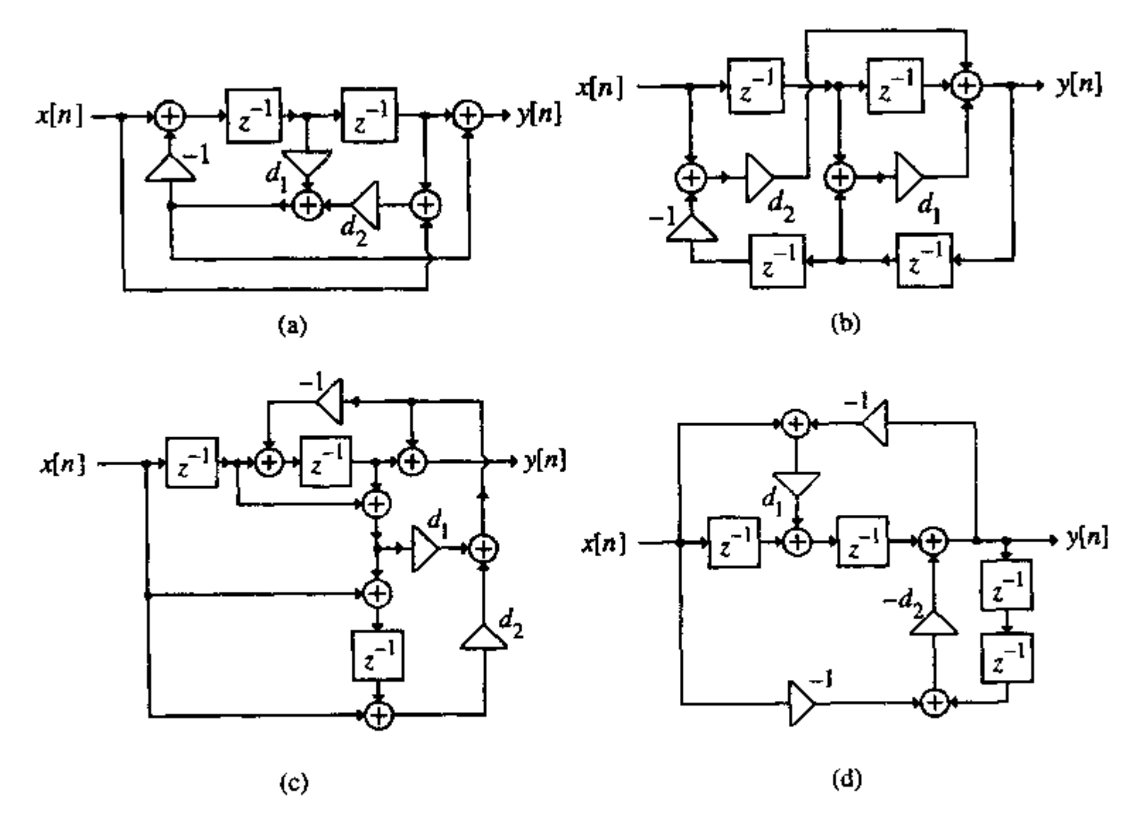


图6.11 (a)3A型全通结构,(b)3D型全通结构,(c)3C型全通结构,(d)3H型全通结构

R6.17 M阶全通传输函数的级联格型实现基于如下方法:由一个m阶全通传输函数$A_m(z)$生成一系列$(m-1)$阶全通传输函数$A_{m-1}(z)$, $m=M,M-1,\cdots,1$[Vai87]:

$$A_m(z)=\frac{d_m+d_{m-1}z^{-1}+d_{m-2}z^{-2}+\ldots+d_1z^{-(m-1)}+z^{-m}}{1+d_1z^{-1}+d_2z^{-2}+\ldots+d_{m-1}z^{-(m-1)}+d_mz^{-m}} \tag{6.16}$$

使用递归

$$A_{m-1}(z)=z\left[\frac{A_m(z)-k_m}{1-k_mA_m(z)}\right], \quad m=M,M-1,\cdots,1, \tag{6.17}$$

其中$k_m=A_m(\infty)=d_m$。当且仅当

$$k_m^2<1, \quad m=M,M-1,\cdots,1 \tag{6.18}$$

时,$A_M(z)$是稳定的。若全通传输函数$A_{m-1}(z)$表示为

$$A_{m-1}(z)=\frac{d'_{m-1}+d'_{m-2}z^{-1}+\ldots+d'_1z^{-(m-2)}+z^{-(m-1)}}{1+d'_1z^{-1}+\ldots+d'_{m-2}z^{-(m-2)}+d'_{m-1}z^{-(m-1)}} \tag{6.19}$$

则$A_{m-1}(z)$的系数可通过表达式

$$d'_i=\frac{d_i-d_md_{m-i}}{1-d_m^2}, \quad i=m-1,m-2,\cdots,2,1 \tag{6.20}$$

与 $A_m(z)$的系数简单地关联起来。图6.12(a)显示了基于式(6.17)的递归的 $A_m(z)$的一个实现。图6.12(b)则显示了基于这种递归的 $A_M(z)$的级联格型实现。

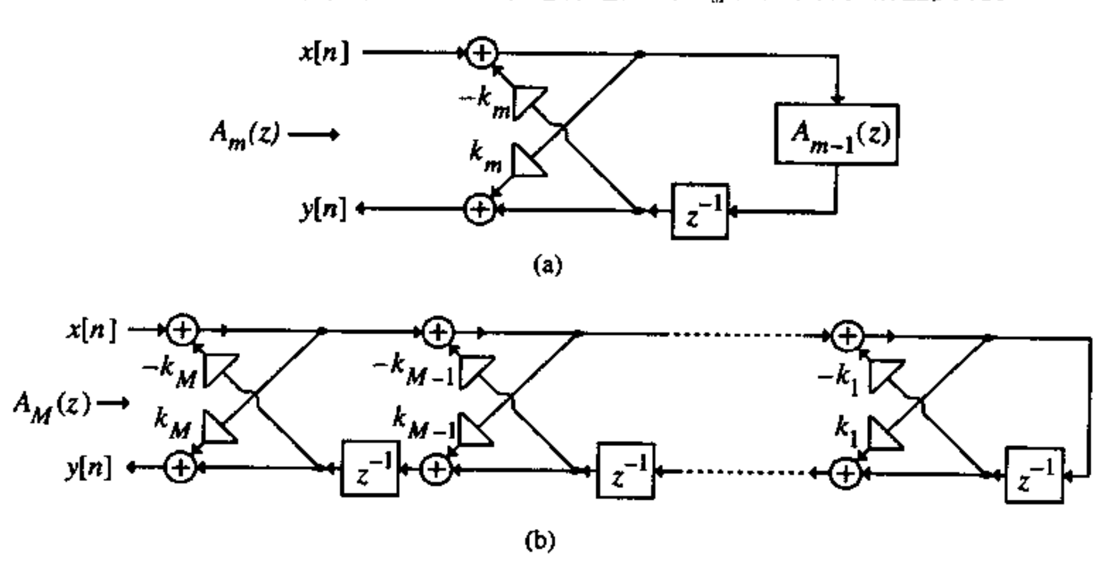


图6.12　(a)$A_m(z)$的实现,(b)$A_M(z)$的级联实现

R6.18 对于任意 M 阶传输函数 $H(z)$的实现,图6.12所示级联格型结构形成了 Gray-Markel 方法的基础[Gra73]。在该方法中,$H(z)=P_M(z)/D_M(z)$用两步实现。在第一步中,中间的全通传输函数 $A_M(z)=z^{-M}D_M(z^{-1})/D_M(z)$以级联格型结构实现。该结构的状态变量在第二步中用适当的权重求和来产生所需的分子 $P_M(z)$。

为了说明实现分子的方法,先简化考虑如下三阶无限冲激响应传输函数的实现:

$$H(z)=\frac{P_3(z)}{D_3(z)}=\frac{p_0+p_1z^{-1}+p_2z^{-2}+p_3z^{-3}}{1+d_1z^{-1}+d_2z^{-2}+d_3z^{-3}} \tag{6.21}$$

为此,首先实现全通函数 $A_3(z)=Y_1(z)/X_1(z)=z^{-3}D_3(z^{-1})/D_3(z)$,如图6.13(a)所示,其中

$$\begin{aligned} d_1' &= \frac{d_1-d_3d_2}{1-d_3^2} \\ d_2' &= \frac{d_2-d_3d_1}{1-d_3^2} \\ d_1'' &= \frac{d_1'-d_2'd_1'}{1-(d_2')^2}=\frac{d_1'}{1+d_2'} \end{aligned} \tag{6.22}$$

接下来,对信号变量 Y_1,S_1,S_2 和 S_3 以权重$\{\alpha_i\}$求和,如图6.13(b)所示,以得到所求的分子 $P_3(z)$。权重$\{\alpha_i\}$为

$$\begin{aligned} \alpha_1 &= p_3 \\ \alpha_2 &= p_2-\alpha_1d_1 \\ \alpha_3 &= p_1-\alpha_1d_2-\alpha_2d_1' \\ \alpha_3 &= p_0-\alpha_1d_3-\alpha_2d_2'-\alpha_3d_1'' \end{aligned} \tag{6.23}$$

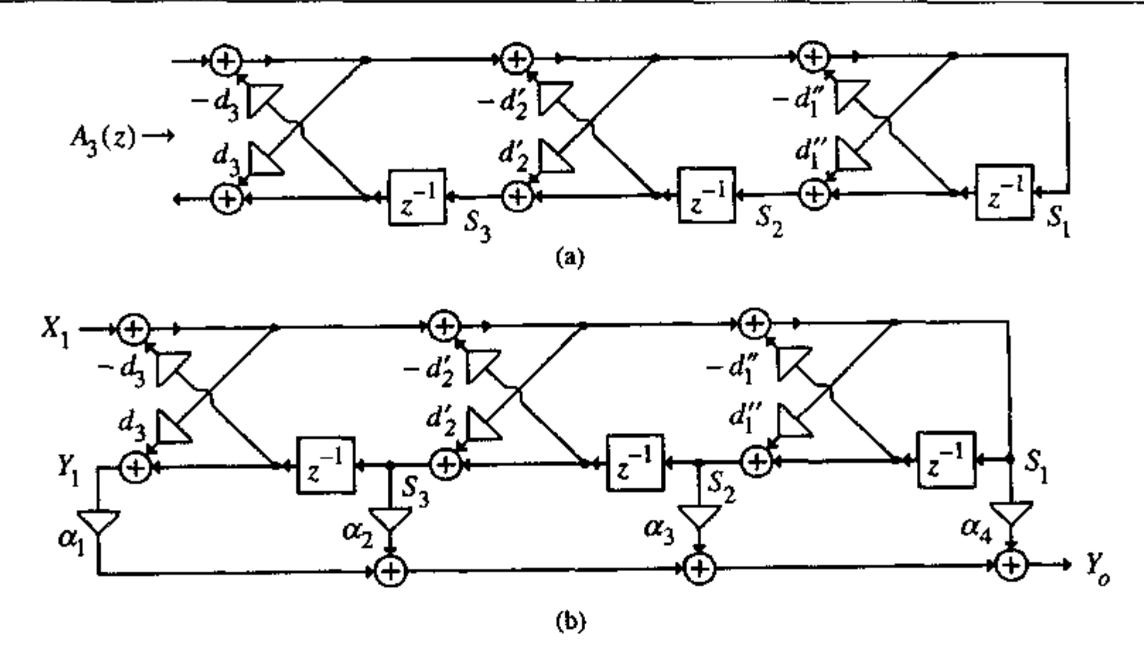


图 6.13 (a)一个三阶全通传输函数的级联格型实现,(b)式(6.21)给出的 $H(z)$的 Gray-Markel 实现

R6.19 设 $G(z)$是一个 N 阶因果有界实无限冲激响应传输函数,其分子是对称的;设 $H(z)$是 $G(z)$的 N 阶功率互补因果有界实传输函数,其分子是反对称的。那么 $G(z)$和 $H(z)$可以按照下面的形式分解:

$$G(z) = \frac{1}{2}\{A_0(z) + A_1(z)\},$$
$$H(z) = \frac{1}{2}\{A_0(z) - A_1(z)\} \tag{6.24}$$

其中,$A_0(z)$和 $A_1(z)$是因果稳定的全通传输函数,其次数之和为 N[Vai86]。基于上述分解的 $G(z)$和 $H(z)$的实现因此如图 6.14 所示。

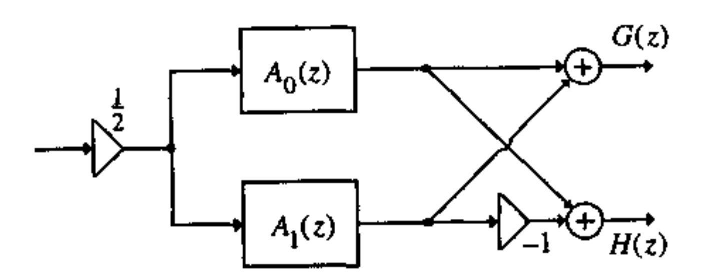


图 6.14 一个无限冲激响应传输函数的并联全通实现

在奇数阶数字巴特沃兹、切比雪夫及椭圆低通或高通数字传输函数的情况中(在第 7 章讨论),有一种由原低通传输函数 $G(z)$或 $H(z)$的极点 $\lambda_k, 0 \leq k \leq N-1$ 来确定全通传输函数$A_0(z)$和 $A_1(z)$的极点的简单方法。设 θ_k 表示极点 λ_k 的角度。若假定极点按 $\theta_k < \theta_{k+1}$ 排列,则 $A_0(z)$的极点由 θ_{2k} 给出,而 $A_1(z)$的极点由 θ_{2k+1} 给出[Gas85]。图 6.15 说明了两个全通传输函数的这种极点交错特性。全通传输函数的零点,位于与极点位置相对的镜像位置。

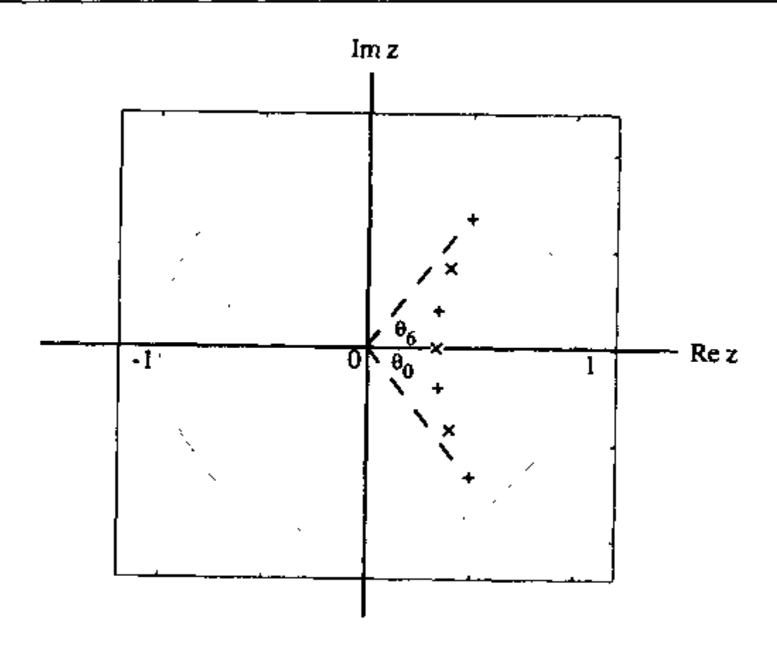


图 6.15 极点交错特性的说明。标记 + 的极点属于 $A_0(z)$,而标记 × 的极点属于 $A_1(z)$

6.3 用到的 MATLAB 命令

本次练习中将要用到的 MATLAB 命令如下:

通用命令

```
disp    length
```

运算符和特殊符号

```
:    .    +    -    *    /    ;    %
```

基本矩阵和矩阵操作

```
ones    pi    :
```

信号处理工具箱

```
latc2tf    poly2rc    residue    residuez    tf2latc    zp2sos
```

关于这些命令的详细说明,请参阅 *MATLAB Reference Guide*[Mat94]和 *Signal Processing Toolbox User's Guide*[Mat96],或在命令窗口中键入 help *命令名称*。附录 B 给出了此处使用的 MATLAB 函数的简要解释。

6.4 有限冲激响应传输函数的实现

例 6.1 级联的实现

式(6.3)给出的 $M-1$ 阶因果有限冲激响应传输函数 $H(z)$ 的因式形式,可由式(6.1)给出的多项式形式表示来确定,该表示因此可用于以级联形式实现 $H(z)$。为此,可使用程序 P6.1

的一种修改形式,程序 P6.1 中使用了函数 zp2sos。

```
% 程序 P6.1
% 将一个有理数传输函数
% 转换为因式形式
num = input('分子系数向量 = ');
den = input('分母系数向量 = ');
[z,p,k] = tf2zp(num,den);
sos = zp2sos(z,p,k)
```

习题

Q6.1 使用程序 P6.1,生成如下有限冲激响应传输函数的一个级联实现:

$$H_1(z) = 2 + 10z^{-1} + 23z^{-2} + 34z^{-3} + 31z^{-4} + 16z^{-5} + 4z^{-6} \tag{6.25}$$

画出级联实现的框图。$H_1(z)$是一个线性相位传输函数吗?

Q6.2 使用程序 P6.1,生成如下有限冲激响应传输函数的一个级联实现:

$$H_2(z) = 6 + 31z^{-1} + 74z^{-2} + 102z^{-3} + 74z^{-4} + 31z^{-5} + 6z^{-6} \tag{6.26}$$

画出级联实现的框图。$H_2(z)$是一个线性相位传输函数吗? 只用 4 个乘法器生成$H_2(z)$的一个级联实现。显示新的级联结构的框图。

6.5 无限冲激响应传输函数的实现

例 6.2 级联和并联实现

式(6.8)中给出的 N 阶因果无限冲激响应传输函数 $H(z)$,其因式形式可由式(6.4)所给出的有理数形式表示确定,然后可用因式形式的级联形式来实现 $H(z)$。为此,可用到程序 P6.1。

习题

Q6.3 使用程序 P6.1 生成如下因果无限冲激响应传输函数的级联实现:

$$H_1(z) = \frac{3 + 8z^{-1} + 12z^{-2} + 7z^{-3} + 2z^{-4} - 2z^{-5}}{16 + 24z^{-1} + 24z^{-2} + 14z^{-3} + 5z^{-4} + z^{-5}} \tag{6.27}$$

画出级联实现的框图。

Q6.4 使用程序 P6.1 生成如下因果无限冲激响应传输函数的级联实现:

$$H_2(z) = \frac{2 + 10z^{-1} + 23z^{-2} + 34z^{-3} + 31z^{-4} + 16z^{-5} + 4z^{-6}}{36 + 78z^{-1} + 87z^{-2} + 59z^{-3} + 26z^{-4} + 7z^{-5} + z^{-6}} \tag{6.28}$$

画出级联实现的框图。

因果无限冲激响应传输函数有两种并联形式实现。并联 I 型基于如式(6.10)所示的 z^{-1} 的部分分式展开,它可用 MATLAB 函数 residuez 得到。并联 II 型基于如式(6.11)所示的 z 的部分分式展开,它可用函数 residue 得到。程序 P6.2 生成两种类型的并联实现。

```
%程序 P6.2
% 一个无限冲激响应传输函数的并联形式实现
num = input('分子系数向量 = ');
den = input('分母系数向量 = ');
[r1,p1,k1] = residuez(num,den);
[r2,p2,k2] = residue(num,den);
disp('并联 I 型')
disp('留数是');disp(r1);
disp('极点在');disp(p1);
disp('常数');disp(k1);
disp('并联 II 型')
disp('留数是');disp(r2);
disp('极点在');disp(p2);
disp('常数');disp(k2);
```

习题

Q6.5 使用程序 P6.2 生成式(6.27)所示因果无限冲激响应传输函数的两种不同并联形式实现。画出两种实现的框图。

Q6.6 使用程序 P6.2 生成式(6.28)所示因果无限冲激响应传输函数的两种不同并联形式实现。画出两种实现的框图。

例 6.3　全通传输函数的实现

M 阶因果无限冲激响应全通传输函数 $A_M(z)$ 的级联格型实现基于 R6.17 中列出的递归算法。格型参数 $\{k_i\}$ 可在 MATLAB 中用函数 poly2rc 确定。为此,也可用到程序 P4.4。

习题

Q6.7 使用程序 P4.4 生成如下全通传输函数的级联格型实现:

$$A_5(z)=\frac{1+5z^{-1}+14z^{-2}+24z^{-3}+24z^{-4}+16z^{-5}}{16+24z^{-1}+24z^{-2}+14z^{-3}+5z^{-4}+z^{-5}} \tag{6.29}$$

$A_5(z)$ 是一个稳定的传输函数吗?

Q6.8 使用程序 P4.4 生成如下全通传输函数的级联格型实现:

$$A_6(z)=\frac{1+7z^{-1}+26z^{-2}+59z^{-3}+87z^{-4}+78z^{-5}+36z^{-6}}{36+78z^{-1}+87z^{-2}+59z^{-3}+26z^{-4}+7z^{-5}+z^{-6}} \tag{6.30}$$

$A_6(z)$ 是一个稳定的传输函数吗?

高阶全通传输函数也可用程序 P6.1 的一种修改形式,按照 R6.13 ~ R6.15 描述的二阶和一阶全通部分的级联来实现。注意,经由 zp2sos 得到的分子和分母因式对不会导致全通部分。然而,通过形成分子分解的镜像因式,可很容易地确定分母因数。

习题

Q6.9 使用 1 型和 2 型全通项生成式(6.29)所示全通传输函数的典范级联实现。显示实现的框图。在最终的结构中,乘法器的总数是多少?

习题

Q6.13 使用函数 `tf2latc` 编写出一个 MATLAB 程序，以生成一个因果无限冲激响应传输函数的 Gray-Markel 实现。用该程序实现式(6.27)所示的传输函数。你的结果与习题 6.11 中得到的结果相符吗？使用函数 `latc2tf` 由向量 `k` 和 `alpha` 确定传输函数。所得到的传输函数和式(6.27)给出的传输函数相同吗？

Q6.14 使用在习题 Q6.13 中生成的程序，实现式(6.28)给出的传输函数。你的结果与习题 6.12 中得到的结果相符吗？使用函数 `latc2tf` 由向量 `k` 和 `alpha` 确定传输函数。所得到的传输函数和式(6.28)给出的传输函数相同吗？

例 6.5　无限冲激响应传输函数的并联全通实现

习题

Q6.15 生成下式给出的三阶因果有界实低通 1 型切比雪夫传输函数 $G(z)$ 的全通和的分解。

$$G(z)=\frac{0.0736+0.2208\,z^{-1}+0.2208\,z^{-2}+0.0736\,z^{-3}}{1-0.9761\,z^{-1}+0.8568\,z^{-2}-0.2919\,z^{-3}} \tag{6.31}$$

其功率互补传输函数 $H(z)$ 的表达式是什么？两个全通传输函数的阶数是什么？用至多 5 个乘法器生成 $G(z)$ 和 $H(z)$ 的并联全通实现，其中可采用一阶和(或)二阶级联来实现两个全通传输函数。

Q6.16 生成一个五阶因果有界实低通椭圆传输函数 $G(z)$ 的全通和的分解，其中

$$G(z)=\frac{0.0417+0.07675\,z^{-1}+0.1203\,z^{-2}+0.1203\,z^{-3}+0.0767\,z^{-4}+0.0417\,z^{-5}}{1-1.8499\,z^{-1}+2.5153\,z^{-2}-1.9106\,z^{-3}+0.9565\,z^{-4}-0.234\,z^{-5}} \tag{6.32}$$

其功率补充传输函数 $H(z)$ 的表达式是什么？两个全通传输函数的阶数是什么？用至多 5 个乘法器生成 $G(z)$ 和 $H(z)$ 的并联全通实现，其中可采用一阶和(或)二阶级联来实现两个全通传输函数。

6.6　基础知识阅读

[1] A. Antoniou. *Digital Filters: Analysis, Design, and Applications*. McGraw-Hill, New York NY, second edition, 1993. Ch. 4.

[2] E. Cunningham. *Digital Filtering: An Introduction*. Houghton-Mifflin, Boston MA, 1992. Sec. 3.12.

[3] D.J. DeFatta, J.G. Lucas, and W.S. Hodgkiss. *Digital Signal Processing: A System Design Approach*. Wiley, New York NY, 1988. Sec. 3.5.

[4] L.B. Jackson. *Digital Filters and Signal Processing*. Kluwer, Boston MA, third edition, 1996. Ch. 5.

[5] R. Kuc. *Introduction to Digital Signal Processing*. McGraw-Hill, New York NY, 1988. Ch. 6.

[6] L.C. Ludeman. *Fundamentals of Digital Signal Processing*. Harper & Row, New York NY, 1986. Ch. 5.

[7] S.K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, New York NY, 1998. Ch. 6.

[8] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs NJ, 1989. Secs. 6.1 – 6.6.

[9] S.J. Orfanidis. *Introduction to Signal Processing*. Prentice-Hall, Englewood Cliffs NJ, 1996. Secs. 4.2.2, 7.1 – 7.3.

[10] B. Porat. *A Course in Digital Signal Procesing*. Wiley, New York NY, 1996. Sec. 11.1.

[11] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs NJ, third edition, 1996. Secs. 7.1 – 7.3.

[12] R.A. Roberts and C.T. Mullis. *Digital Signal Processing*. Addison-Wesley, Reading MA, 1987. Sec. 3.7.

第 7 章　数字滤波器设计

7.1　导言

设计传输函数 $G(z)$，使其频率响应 $G(e^{j\omega})$ 逼近于给定的频率响应指标，这个过程称为数字滤波器设计。在得到 $G(z)$ 之后，就以一种适当的滤波器结构的形式实现它。在前面的实验练习中，已考虑过有限冲激响应和无限冲激响应传输函数的实现。在这个实验练习中，我们将学会设计满足某个给定幅度或增益响应的无限冲激响应或有限冲激响应数字滤波器。

7.2　基础知识回顾

R7.1　滤波器指标通常按照其幅度响应指定。例如，低通滤波器 $G(z)$ 的幅度响应 $|G(e^{j\omega})|$ 通常如图 7.1 所示。在由 $0\leqslant\omega\leqslant\omega_p$ 所定义的通带中，我们要求

$$1-\delta_p\leqslant|G(e^{j\omega})|\leqslant 1+\delta_p,\qquad |\omega|\leqslant\omega_p \tag{7.1}$$

换句话说，幅度以误差 $\pm\delta_p$ 逼近 1。在由 $\omega_s\leqslant|\omega|\leqslant\pi$ 定义的阻带中，我们要求

$$|G(e^{j\omega})|\leqslant\delta_s,\qquad \omega_s\leqslant|\omega|\leqslant\pi \tag{7.2}$$

即幅度以误差 δ_s 接近于零。频率 ω_p 和 ω_s 分别称为通带边界频率和阻带边界频率。在通带和阻带内的最大容限，δ_p 和 δ_s，称为波纹。

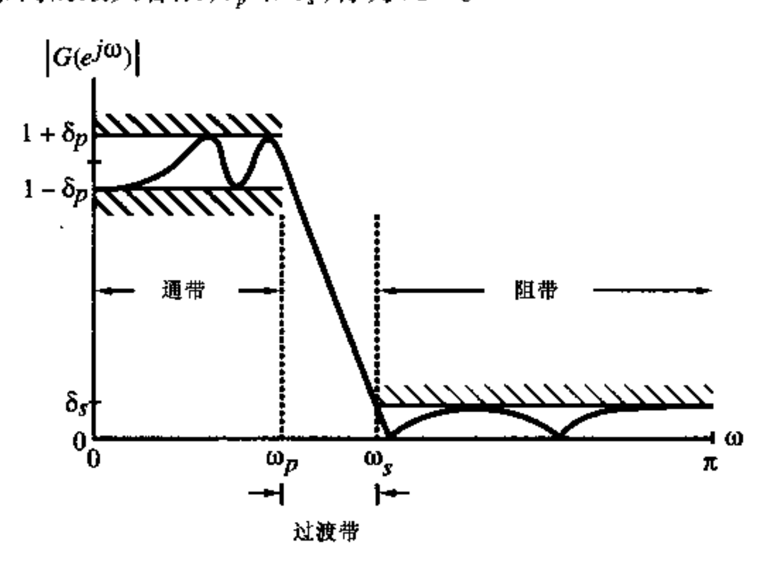


图 7.1　一个数字低通滤波器的典型幅度响应指标

R7.2　在大多数应用中，给定的数字滤波器指标如图 7.2 所示。此时，在由 $0\leqslant\omega\leqslant\omega_p$ 定义的通带中，幅度的最大值和最小值分别是 1 和 $1/\sqrt{1+\varepsilon^2}$。单位为 dB 的峰通带波纹是

$$R_p = 20\log_{10}\sqrt{1+\epsilon^2}\ \text{dB} \tag{7.3}$$

由 $\omega_s \leqslant |\omega| \leqslant \pi$ 定义的阻带中，最大波纹由 $1/A$ 表示；单位为 dB 的最小阻带衰减为

$$R_s = 20\log_{10} A\ \text{dB} \tag{7.4}$$

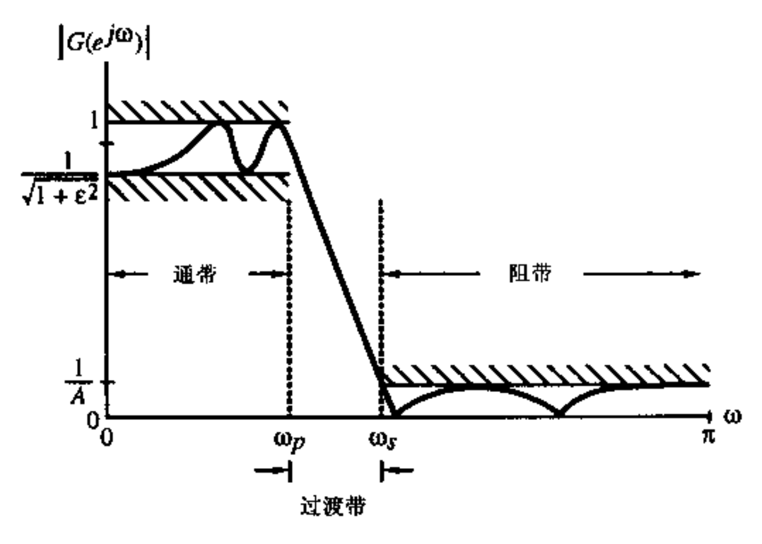


图 7.2 一个数字低通滤波器的归一化幅度响应指标

R7.3 若通带边界频率 F_p和阻带边界频率 F_s与数字滤波器的抽样率 F_T 一起以 Hz 为单位给定，则以弧度为单位的归一化角边界频率为

$$\omega_p = \frac{\Omega_p}{F_T} = \frac{2\pi F_p}{F_T} = 2\pi F_p T \tag{7.5}$$

$$\omega_s = \frac{\Omega_s}{F_T} = \frac{2\pi F_s}{F_T} = 2\pi F_s T \tag{7.6}$$

R7.4 滤波器设计过程的第一步是估计传输函数阶数。对于基于模拟低通滤波器 $H_a(s)$的转换的无限冲激响应数字低通滤波器 $G(z)$的设计，存在估计滤波器阶数的一个解析公式。对于有限冲激响应低通或高通数字滤波器的设计，可以通过几种设计公式，由下列数字滤波器指标来直接估计最小滤波器长度 N：归一化通带边界角频率 ω_p，归一化阻带边界角频率 ω_s，峰通带波纹 δ_p，峰阻带波纹 δ_s。由凯泽[Kai74]生成的一个相当简单的逼近式为

$$N \approx \frac{-20\log_{10}(\sqrt{\delta_p\delta_s}) - 13}{14.6(\Delta\omega)/2\pi} \tag{7.7}$$

其中 $\Delta\omega = |\omega_p - \omega_s|$是过渡带的宽度。上述公式也可用于设计多过渡带有限冲激响应滤波器，在这种情况下，$\Delta\omega$ 是所有过渡带的最小宽度。对于具有不等过渡带的多频带滤波器，用上述公式设计的滤波器可能在那些较宽的过渡带中表现出令人无法接受的幅度响应，在这种情况下，这些频带将被处理得较小，直到得到一个可接受的幅度响应。

R7.5 由 Herrmann，Rabiner 和 Chan[Her73]给出的一个稍微精确些的长度值为

$$N \approx \frac{D_\infty(\delta_p,\delta_s) - F(\delta_p,\delta_s)\left[(\omega_s-\omega_p)/2\pi\right]^2}{\left[(\omega_s-\omega_p)/2\pi\right]} \tag{7.8}$$

其中，

$$D_\infty(\delta_p,\delta_s) = \left[a_1(\log_{10}\delta_p)^2 + a_2(\log_{10}\delta_p) + a_3\right]\log_{10}\delta_s - \left[a_4(\log_{10}\delta_p)^2 + a_5(\log_{10}\delta_p) + a_6\right] \tag{7.9}$$

且

$$F(\delta_p,\delta_s) = b_1 + b_2\left[\log_{10}\delta_p - log_{10}\delta_s\right] \tag{7.10}$$

并有

$$\begin{aligned} a_1 &= 0.005309, \quad a_2 = 0.07114, \quad a_3 = -0.4761, \\ a_4 &= 0.00266, \quad a_5 = 0.5941, \quad a_6 = 0.4278, \\ b_1 &= 11.01217, \quad b_2 = 0.51244 \end{aligned} \tag{7.11}$$

式(7.8)所给的公式对 $\delta_p > \delta_s$ 有效。若 $\delta_p < \delta_s$，则滤波器长度公式要通过互换式(7.8)中的 δ_p 和 δ_s 得到。对于 δ_p 和 δ_s 的较小值，式(7.7)和式(7.8)均提供相当接近且准确的结果。另一方面，当 δ_p 和 δ_s 的值较大时，式(7.8)会产生一个更精确的长度值。

R7.6 在许多滤波器设计问题中，用式(7.7)或式(7.8)估算的阶数可能得到不满足给定指标的滤波器。在这种情况下，需要逐渐增加 N 值，直到满足给定指标。

R7.7 广泛用于设计无限冲激响应滤波器的方法是基于从 s 平面到 z 平面的双线性变换。即

$$s = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \tag{7.12}$$

使用上面的变换，模拟传输函数 $H_a(s)$ 按照下式被转换成数字传输函数 $G(z)$：

$$G(z) = H_a(s)\big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} \tag{7.13}$$

R7.8 对于双线性变换，s 平面中的虚轴($s = j\Omega$)与 z 平面中的单位圆($z = e^{j\omega}$)间的关系为

$$\Omega = \tan(\omega/2) \tag{7.14}$$

其 s 平面中的整个虚轴映射到 z 平面中的单位圆，并在频率轴中引入了称为畸变的失真。为了生成满足某个给定幅度响应的数字滤波器，首先用式(7.14)给出的关系得到数字滤波器的临界频带边界频率(ω_p 和 ω_s)的模拟等效(Ω_p 和 Ω_s)，然后用预畸变临界频率设计模拟原型 $H_a(s)$，并用双线性变换对 $H_a(s)$ 变换，以得到所求的数字滤波器传输函数 $G(z)$。

R7.9 对理想频率响应 $H_D(e^{j\omega})$ 实行离散时间傅里叶逆变换，可得到其理想无限长冲激响应 $h_D[n]$。有限冲激响应滤波器设计的最直接方法，是用一个合适的有限长窗函数 $w[n]$ 对 $h_D[n]$ 加窗。最终设计的冲激响应系数则为 $h[n] = h_D[n]\cdot w[n]$。

R7.10 图 4.1(a)所示理想低通滤波器有一个零相位频率响应

$$H_{LP}(e^{j\omega}) = \begin{cases} 1, & |\omega| \leqslant \omega_c \\ 0, & \omega_c < |\omega| \leqslant \pi \end{cases} \tag{7.15}$$

相应的冲激响应系数 $h_{LP}[n]$ 为

$$h_{LP}[n] = \frac{\sin \omega_c n}{\pi n}, \quad -\infty \leqslant n \leqslant \infty \tag{7.16}$$

显然它是无限的且非绝对可和的，因此不可实现。通过将范围 $-M \leqslant n \leqslant M$ 之外的所

有冲激响应系数设为零，可得到长度为 $N=2M+1$ 的有限长非因果逼近，当它向右移动时，可得到因果有限冲激响应低通滤波器的系数：

$$\hat{h}_{LP}[n]=\begin{cases}\frac{\sin\omega_c(n-M)}{\pi(n-M)}, & 0\leqslant n\leqslant N-1\\ 0, & \text{其他}\end{cases} \tag{7.17}$$

注意，当 N 为偶数时，上式也成立。此时 M 是一个小数。

图 4.1(b)所示的理想高通滤波器的冲激响应系数 $h_{HP}[n]$ 为

$$h_{HP}[n]=\begin{cases}1-\frac{\omega_c}{\pi}, & n=0,\\ -\frac{\sin(\omega_c n)}{\pi n}, & |n|>0\end{cases} \tag{7.18}$$

截止频率在 ω_{c1} 和 ω_{c2} 的、的图 4.1(c)所示的理想带通滤波器的冲激响应系数 $h_{BP}[n]$ 为

$$h_{BP}[n]=\frac{\sin\omega_{c2}n}{\pi n}-\frac{\sin\omega_{c1}n}{\pi n},\quad -\infty\leqslant n\leqslant\infty \tag{7.19}$$

截止频率在 ω_{c1} 和 ω_{c2} 的、如图 4.1(d)所示的理想带阻滤波器的冲激响应系数 $h_{BS}[n]$ 为

$$h_{BS}[n]=\begin{cases}1-\frac{(\omega_{c2}-\omega_{c1})}{\pi}, & n=0\\ \frac{\sin(\omega_{c1}n)}{\pi n}-\frac{\sin(\omega_{c2}n)}{\pi n}, & |n|>0\end{cases} \tag{7.20}$$

R7.11 上述所有设计方法都针对于具有两种幅度级的单通带或单阻带滤波器，但我们可将这些方法直接推广到多级有限冲激响应滤波器的设计中，以得到冲激响应系数的表达式。理想 L 频带数字滤波器 $H_{ML}(z)$ 的零相位频率响应为

$$H_{ML}(e^{j\omega})=A_k,\quad \omega_{k-1}\leqslant\omega\leqslant\omega_k,\quad k=1,2,\cdots,L \tag{7.21}$$

其中 $\omega_0=0$ 而 $\omega_l=\pi$。图 7.3 显示了一个典型的多级滤波器的频率响应。其冲激响应 $h_{ML}[n]$ 为

$$h_{ML}[n]=\sum_{\ell=1}^{L}(A_\ell-A_{\ell+1})\cdot\frac{\sin(\omega_c n)}{\pi n} \tag{7.22}$$

其中 $A_{L+1}=0$。

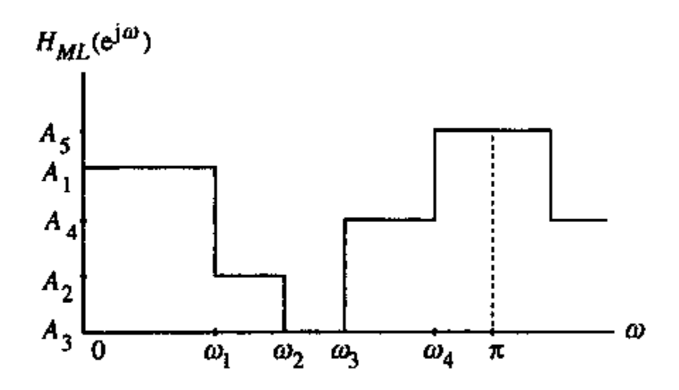


图 7.3 典型的零相位多级频率响应

R7.12 理想希尔伯特变换器，也称为 90 度移相器，由频率响应

$$H_{HT}(e^{j\omega}) = \begin{cases} j, & -\pi < \omega < 0 \\ -j, & 0 < \omega < \pi \end{cases} \tag{7.23}$$

描述。相应的冲激响应 $h_{HT}[n]$为

$$h_{HT}[n] = \begin{cases} 0, & n = 0 \\ \frac{2\sin^2(\pi n/2)}{\pi n}, & n \neq 0 \end{cases} \tag{7.24}$$

R7.13 理想离散时间微分器用频率响应

$$H_{DIF}(e^{j\omega}) = j\omega, \quad |\omega| < \pi \tag{7.25}$$

描述。相应的冲激响应 $h_{DIF}[n]$为

$$h_{DIF}[n] = \begin{cases} 0, & n = 0 \\ \frac{\cos \pi n}{n}, & |n| > 0 \end{cases} \tag{7.26}$$

与理想低通滤波器一样，上面所有的理想滤波器[式(7.18)~式(7.20)、式(7.22)、式(7.24)和式(7.26)]是不可实现的。它们可通过截短冲激响应序列到有限长，并向右适当地移动截短的系数来实现。

R7.14 通过简单截短前一节给出的理想滤波器的冲激响应系数所得到的因果有限冲激响应滤波器，在它们各自的幅度响应中会出现一种摆动行为，它们通常称为吉布斯现象。吉布斯现象可用如下方式来减弱：用一个两端平滑地降为零的窗口来加窗，或提供一个从通带到阻带平滑的转换。使用渐进窗口会引起旁瓣高度的减小和相应主瓣宽度的增加，得到不连续点处的一个较宽过渡。在所有基于窗口的低通滤波器设计中，截止频率 ω_c 是通带和阻带边界频率总和的一半。

R7.15 具有固定波纹且长度为 $2M+1$ 的常用渐进窗是

汉宁：$$w[n] = \frac{1}{2}\left[1 + \cos\left(\frac{2\pi n}{2M+1}\right)\right], \quad -M \leqslant n \leqslant M \tag{7.27}$$

海明：$$w[n] = 0.54 + 0.46\cos\left(\frac{2\pi n}{2M+1}\right), \quad -M \leqslant n \leqslant M \tag{7.28}$$

布莱克曼：$$w[n] = 0.42 + 0.5\cos\left(\frac{2\pi n}{2M+1}\right) + 0.08\cos\left(\frac{4\pi n}{2M+1}\right), \quad -M \leqslant n \leqslant M \tag{7.29}$$

R7.16 长度 $2M+1$ 的多尔夫-切比雪夫窗是一个可调窗，定义为

$$w[n] = \frac{1}{2M+1}\left[\frac{1}{\gamma} + 2\sum_{k=1}^{M} T_k\left(\beta \cos\frac{k\pi}{2M+1}\right)\cos\frac{2nk\pi}{2M+1}\right] \quad -M \leqslant n \leqslant M \tag{7.30}$$

其中，

$$\gamma = \frac{\text{旁瓣振幅}}{\text{主瓣振幅)}} \tag{7.31}$$

$$\beta = \cosh\left(\frac{1}{2M}\cosh^{-1}\frac{1}{\gamma}\right) \tag{7.32}$$

k 阶切比雪夫多项式 $T_k(x)$ 定义为

$$T_k(x)=\begin{cases}\cos(k\cos^{-1}x), & \text{for } |x|\leqslant 1\\ \cosh(k\cosh^{-1}x), & \text{for } |x|>1\end{cases} \tag{7.33}$$

R7.17 使用最广泛的可调窗是凯泽窗，为

$$w[n]=\frac{I_0(\beta\sqrt{1-(n/M)^2})}{I_0(\beta)},\quad -M\leqslant n\leqslant M \tag{7.34}$$

其中，β 是一种可调整的参数，而 $I_0(u)$ 是修正的零阶贝塞尔函数，它可用幂级数形式表示为

$$I_0(u)=1+\sum_{r=1}^{\infty}\left[\frac{(u/2)^r}{r!}\right]^2 \tag{7.35}$$

可以看出，$I_0(u)$ 对 u 的所有实数值是正数。实际上，在式(7.35)的和中只保留前20项就足以得到相当精确的 $I_0(u)$ 值。参数 β 控制加窗的滤波器响应的最小阻带衰减 $\alpha_s=-20\log_{10}\delta_s$。对于给定的 α_s 和过渡带宽 $\Delta f=F_p-F_s$，估算 β 和滤波器长度 $N=2M+1$ 的公式为

$$\beta=\begin{cases}0.1102(\alpha_s-8.7), & \alpha_s>50\\ 0.5842(\alpha_s-21)^{0.4}+0.078\,86(\alpha_s-21), & 21\leqslant\alpha_s\leqslant 50\\ 0, & \alpha_s<21\end{cases} \tag{7.36}$$

及

$$N=\begin{cases}\frac{\alpha_s-7.95}{14.36\,\Delta f}+1, & \alpha_s>21\\ \frac{0.9222}{\Delta f}+1, & \alpha_s\leqslant 21\end{cases} \tag{7.37}$$

凯泽窗在通带波纹 δ_p 上提供非独立的控制。但实际上，δ_p 近似等于 δ_s。

R7.18 通过最小化权重误差 ε 的峰绝对值，得到的线性相位有限冲激响应滤波器为

$$\varepsilon=\max_{0\leqslant\omega\leqslant\pi}|\mathcal{E}(\omega)| \tag{7.38}$$

它通常称为最优有限冲激响应滤波器，其中权重误差函数 $\mathcal{E}(\omega)$ 定义为

$$\mathcal{E}(\omega)=P(\omega)\left[|H(e^{j\omega})|-D(\omega)\right] \tag{7.39}$$

在所关心的频率范围中，其表现出等波纹行为。广泛用来设计最优线性相位有限冲激响应滤波器的有效算法是Parks-McClellan算法[Par72]。

7.3 用到的MATLAB命令

这个练习中将用到的MATLAB命令如下：

通用命令

```
disp    length
```

运算符和特殊符号

```
:    .    +    -    *    /    ;
%    .*   ./   >    =    ==
```

语言构造和调试

```
else    function    if
```

基本矩阵和矩阵控制

```
fliplr    nargin    pi    :
```

基本函数

```
abs    ceil    cos    log10    sin    sqrt
```

数据分析

```
min
```

二维图形

```
axis    grid    plot    title    xlabel    ylabel
```

信号处理工具箱

```
blackman    butter    buttord     chebwin    cheb1ord
cheb2ord    cheb11    cheby2      ellip      ellipord
fir1        fir2      freqz       hanning    hamming
kaiser      remez     remezord
```

关于这些命令的详细说明，请参阅 *MATLAB Reference Guide*[Mat94]和 *Signal Processing Toolbox User's Guide*[Mat96]，或在命令窗口中键入 help *命令名称*。附录 B 给出了此处使用的 MATLAB 函数的简要解释。

7.4　无限冲激响应滤波器设计

最常用于设计无限冲激响应滤波器的方法基于原型模拟传输函数的双线性变换。模拟传输函数通常是下列类型之一：巴特沃兹、切比雪夫 1 型、切比雪夫 2 型和椭圆传输函数。这些滤波器类型之间的差别可通过考虑模拟低通滤波器来说明。巴特沃兹低通传输函数在 dc(即 $\Omega = 0$)处有一个最大平坦的幅度响应，且随着频率增加幅度响应单调地减少。切比雪夫 1 型低通传输函数在通带中有一个等波纹幅度响应，并且在通带外随着频率增加幅度响应单调地减少。切比雪夫 2 型低通传输函数在通带中有一个随着频率增加单调减少的幅度响应，并且在通带中有一个等波纹的幅度响应。椭圆传输函数在通带中和在阻带中均有一个等波纹幅度响应。

例 7.1　无限冲激响应滤波器的阶数的估计

滤波器设计过程中的第一步是，选择接近所使用的滤波器的类型，然后由滤波器指标来估计传输函数的阶数。用来估计巴特沃兹滤波器的阶数的 MATLAB 命令是

```
[N,Wn] = buttord(Wp,Ws,Rp,Rs)
```

其中输入参数是归一化通带边界频率 Wp、归一化阻带边界频率 Ws、单位为 dB 的通带波

纹 Rp 和单位为 dB 的最小阻带衰减 Rs。由于抽样频率被假定为 2Hz,Wp 和 Ws 均必须是 0 和 1 之间的一个数。输出数据是满足指标的最低阶数 N 和归一化截止频率 Wn 若 Rp = 3 dB,则 Wn = Wp。buttord 也可用于估计高通、带通和带阻巴特沃兹滤波器的阶数。对于高通滤波器设计,Wp > Ws。对于带通和带阻滤波器设计,Wp 和 Ws 是指定边界频率的双元素向量,其中较低的边界频率是向量的第一个元素。在后面的情况中,Wn 也是一个双元素向量。

用来估计切比雪夫 1 型滤波器的阶数的 MATLAB 命令是

```
[N,Wn] = cheb1ord(Wp,Ws,Rp,Rs)
```

用来设计切比雪夫 2 型滤波器的阶数的 MATLAB 命令是

```
[N,Wn] = cheb2ord(Wp,Ws,Rp,Rs)
```

用来设计椭圆滤波器的阶数的 MATLAB 命令是

```
[N,Wn] = ellipord(Wp,Ws,Rp,Rs)
```

和前面一样,Wp 和 Ws 是值在 0 和 1 之间的通带和阻带边界频率。同样,Rp 和 Rs 是单位为 dB 的最小阻带衰减。N 包含估算出的最低阶数,而 Wn 是截止频率。注意,对于带通和带阻滤波器设计,使用适当滤波器设计命令得到的传输函数的实际阶次是 2N。

习题

Q7.1 用 MATLAB 确定一个数字无限冲激响应低通滤波器所有四种类型的最低阶数。指标如下:40 kHz 的抽样率,4 kHz 的通带边界频率,8 kHz 的阻带边界频率,0.5 dB 的通带波纹,40 dB 的最小阻带衰减。评论你的结果。

Q7.2 用 MATLAB 确定一个数字无限冲激响应高通滤波器所有四种类型的最低阶数。指标如下:3500 Hz 的抽样率,1050 Hz 的通带边界频率,600 Hz 的阻带边界频率,1 dB 的通带波纹,50 dB 的最小阻带衰减。评论你的结果。

Q7.3 用 MATLAB 确定一个数字无限冲激响应带通滤波器所有四种类型的最低阶数。指标如下:7 kHz 的抽样率,1.4 kHz 和 2.1 kHz 的通带边界频率,1.05 kHz 和 2.45 kHz 的阻带边界频率,0.4 dB 的通带波纹,50 dB 的最小阻带衰减。评论你的结果。

Q7.4 用 MATLAB 确定一个数字无限冲激响应带阻滤波器所有四种类型的最低阶数。指标如下:12 kHz 的抽样率,2.1 kHz 和 4.5 kHz 的通带边界频率,2.7 kHz 和 3.9 kHz 的阻带边界频率,0.6 dB 的通带波纹,45 dB 的最小阻带衰减。评论你的结果。

例 7.2 无限冲激响应滤波器设计

在选择了滤波器类型并估计了其阶数后,下一步是确定滤波器的传输函数。为此,MATLAB 对所有四种类型的滤波器提供了函数。设计巴特沃兹数字低通或带通滤波器的 MATLAB 命令是

```
[num,den] = butter(N,Wn)
```

其中输入参数 N 和 Wn 可使用函数 buttord 确定,输出是向量 num 和 den,它们分别是以 z^{-1} 的升幂排列的传输函数的分子和分母多项式的系数。若 Wn 是一个标量,butter 返回一个 N

阶的低通传输函数,若 Wn 是一个双元素向量,它返回一个 2N 阶的带通传输函数。设计 N 阶巴特沃兹数字高通滤波器的 MATLAB 命令是

```
[num,den] = butter(N,Wn,'high')
```

然而,若 Wn 是一个双元素向量,命令

```
[num,den] = butter(N,Wn,'stop')
```

返回一个 2N 阶巴特沃兹带阻滤波器的传输函数。设计切比雪夫 1 型数字滤波器的 MATLAB 命令是

```
[num,den] = cheby1(N,Rp,Wn)
[num,den] = cheby1(N,Rp,Wn,'filtertype')
```

设计切比雪夫 2 型数字滤波器的 MATLAB 命令是

```
[num,den] = cheby2(N,Rs,Wn)
[num,den] = cheby2(N,Rs,Wn,'filtertype')
```

设计椭圆数字滤波器的 MATLAB 命令是

```
[num,den] = ellip(N,Rp,Rs,Wn)
[num,den] = ellip(N,Rp,Rs,Wn,'filtertype')
```

若 Wn 是一个标量,则每种情况下返回的是一个 N 阶低通传输函数,而若 Wn 是一个双元素向量,则返回的是一个 2N 阶带通传输函数。在上面的每一个命令中,设计高通滤波器时,filtertype 是 high,同时 Wn 是一个标量;而设计带阻滤波器时,filtertype 是 stop,同时 Wn 是一个双元素向量。

程序 P7.1 说明巴特沃兹带阻滤波器的设计。

```
%程序 P7.1
%巴特沃兹带阻滤波器的设计
Ws = [0.4 0.6]; Wp = [0.3 0.7]; Rp = 0.4; Rs = 50;
%估计滤波器阶数
[N1,Wn1] = buttord(Wp,Ws,Rp,Rs);
%设计滤波器
[num,den] = butter(N1,Wn1,'stop');
%显示传输函数
disp('分子系数是');disp(num);
disp('分母系数是');disp(den);
%计算增益响应
[g,w] = gain(num,den);
%绘制增益响应
plot(w/pi,g);grid
axis([0 1 -60 5]);
xlabel('\omega /\pi');
ylabel('增益,dB');
title('巴特沃兹带阻滤波器的增益响应');
```

习题

Q7.5 通过运行程序 P7.1 来设计巴特沃兹带阻滤波器。写出所产生的传输函数的准确表达

式。滤波器的指标是什么？你的设计符合指标吗？使用 MATLAB,计算并绘制滤波器的未畸变的相位响应及群延迟响应。

Q7.6 修改程序 P7.1 来设计符合习题 Q7.1 所给指标的切比雪夫 1 型低通滤波器。写出所产生的传输函数的准确表达式。你的设计符合指标吗？使用 MATLAB,计算并绘制滤波器的未畸变的相位响应及群延迟响应。

Q7.7 修改程序 P7.1 来设计符合习题 Q7.2 所给指标的切比雪夫 2 型高通滤波器。写出所产生的传输函数的准确表达式。你的设计符合指标吗？使用 MATLAB,计算并绘制滤波器的未畸变的相位响应及群延迟响应。

Q7.8 修改程序 P7.1 来设计符合习题 Q7.3 所给指标的椭圆带通滤波器。写出所产生的传输函数的准确表达式。你的设计符合指标吗？使用 MATLAB,计算并绘制滤波器的未畸变的相位响应及群延迟响应。

7.5 有限冲激响应滤波器设计

有限冲激响应滤波器设计的最简单方法是,对期望的理想频率响应进行离散时间傅里叶逆变换,得到双无限长冲激响应,对所得的双无限系数简单地截尾为有限系数。然而,简单截尾得到的有限冲激响应滤波器的幅度响应中会有一个振荡行为,它通常称为吉布斯现象。

吉布斯现象可用一个合适的有限长窗口函数对双无限长冲激响应系数加窗来减小。MATLAB 函数 fir1 和 fir2 可以用来设计加窗的有限冲激响应数字滤波器。两个函数均产生一个线性相位设计。

函数 fir1 可用于设计常规的低通、高通、带通和带阻线性相位有限冲激响应滤波器。对于抽样频率为 2 Hz 的情况,命令

```
b = fir1(N,Wn)
```

在向量 b 中返回以 z^{-1} 的升幂排列的 N 阶低通或带通滤波器的冲激响应系数。对于低通设计,归一化截止频率由标量 Wn 给定,它是在 0 和 1 之间的一个数。对于带通设计,Wn 是包含指定通带边界的一个双元素向量[Wn1,Wn2],其中 0 < Wn1 < Wn2 < 1。命令

```
b = fir1(N,Wn,'high')
```

其中 N 为一个偶数,用于设计高通滤波器。命令

```
b = fir1(N,Wn,'stop')
```

其中 Wn 是一个双元素向量,用于设计带阻有限冲激响应滤波器。若没有指定,用海明窗作为默认值。命令

```
b = fir1(N,Wn,taper)
```

使用在向量 taper 中给定的长度为 N + 1 的窗系数。然而,窗系数必须事先用一个适当的 MATLAB 函数来产生,如 blackman,hamming,hanning,chebwin 或 kaiser。用到的命令为

```
taper = blackman(N)    taper = hamming(N)    taper = hanning(N)
taper = chebwin(N)     taper = kaiser(N,beta)
```

函数 fir2 可用于设计带有任意形状幅度响应的线性相位有限冲激响应滤波器。在其基本形状中，命令是

```
b = fir2(N,fpts,mval)
```

它在长度为 N + 1 的向量 b 中返回以 z^{-1} 的升幂排列的冲激响应系数。fpts 是给定的频率点向量，在 0 到 1 范围内以递增的顺序排列，其中第一个频率点是 0，而最后的频率点是 1。和前面一样，抽样频率被假定为 2 Hz。mval 是给定频率点的给定幅度值的向量，因此也必须和 fpts 的长度一样，默认使用海明窗。为使用其他窗口，所用到的命令是

```
b = fir2(N,fpts,mval,taper)
```

其中向量 taper 包含给定的窗系数。

一种广泛使用的用于设计线性相位有限冲激响应滤波器的方法是 Parks-McClellan 算法，它得到一种具有在式(7.39)中定义的等波纹权重误差$\mathcal{E}(\omega)$的最优有限冲激响应滤波器。它使用 Remez 最优化算法且在 MATLAB 中以函数 remez 存在。该函数可用于设计任何类型的单频带或多频带滤波器、微分器和希尔伯特变换器。在其基本形式中，命令

```
b = remez(N,fpts,mval)
```

以 z^{-1} 的升幂返回长度为 N + 1 的向量 b，它包含期望的有限冲激响应滤波器的冲激响应系数。fpts 是给定的频率点向量，在 0 到 1 范围内以递增的顺序排列，其中第一个频率点是 0，而最后的频率点是 1。和前面一样，抽样频率被假定为 2 Hz。在指定的频带边界处，有限冲激响应滤波器频率响应的期望幅度由向量 mval 给出，其中元素以等值对给出。在给定的两个连续频率点 f(k)和 f(k + 1)之间，所期望的幅度按下面的规则确定。对于奇数 k，幅度是连接点{mval(k),fpts(k)}和{mval(k + 1),fpts(k + 1)}的一段线段，而对于偶数 k，幅度未指定，频率范围[fpts(k),fpts(k + 1)]是一个过渡或“不在意”区域。向量 fpts 和 mval 必须具有相同的长度且为偶数。图 7.4 说明了如下向量 fpts 和 mval 之间的关系：

```
fpts = [0 0.2 0.4 0.7 0.8 1.0]
mval = [0.5 0.5 1.0 1.0 0.3 0.3]
```

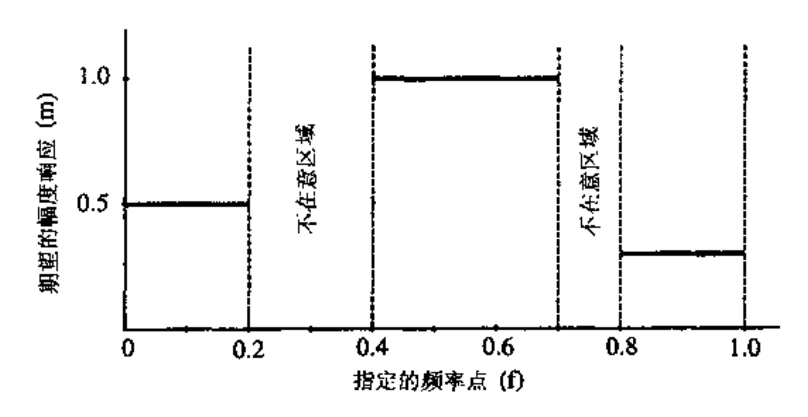


图 7.4 向量 fpts 和 mval 之间关系的说明

在通带和阻带中，所期望的幅度响应可通过一个附加的向量 wgts 来加权，它包含在函数 remez 的自变量中。函数可用于设计等波纹 1 型、2 型、3 型和 4 型线性相位有限冲激响应滤

波器。1 型和 2 型分别是对于阶数 N 为偶数和奇数的默认设计。3 型(N 是偶数)和 4 型(N 是奇数)用于特定的滤波器设计,如希尔伯特变换器和微分器。为了设计这两种类型的有限冲激响应滤波器,在 remez 的后两种形式中,要对 ftype 使用标记 hilbert 和 differentiator。命令

```
b = remez(N,fpts,mval,wgts)
```

用于设计在每一个频带中由权向量 wgts 的元素加权的有限冲激响应滤波器,wgts 的长度因此是 fpts 的一半。向量 wgts 的元素可通过给定的通带和阻带波纹用波纹值除以最大波纹值确定。为设计希尔伯特变换器和微分器,可使用形式

```
remez(N,fpts,mval,ftype)
remez(N,fpts,mval,wgts,ftype)
```

其中 ftype 是字符串 hilbert 或 differentiator。在希尔伯特变换器设计的情况中,fpts 中的最小元素不应是 0。

满足所给指标的有限冲激响应滤波器的阶数 N,可用式(7.7)所示的凯泽公式或式(7.8)所示的 Herrmann 公式估算。下面给出的 MATLAB 函数 kaiord 可实现凯泽方程:

```
function N = kaiord(Fp,Fs,dp,ds,FT)
% 使用凯泽公式计算有限冲激响应
% 多频带滤波器线性相位的长度
% dp 是通带波纹
% ds 是阻带波纹
% Fp 是单位为 Hz 的通带边界
% Fs 是单位为 Hz 的阻带边界
% FT 是单位为 Hz 的抽样频率
% 若未指定,默认值是 2
% N 是估计出的有限冲激响应滤波器的阶数
if nargin = = 4,
    FT = 2;
end
if length(Fp) > 1,
    TBW = min(abs(Fp(1) - Fs(1)),abs(Fp(2) - Fs(2)));
    else
    TBW = abs(Fp - Fs);
end
num = - 20 * log10(sqrt(dp * ds)) - 13;
den = 14.6 * TBW/FT;
N = ceil(num/den);
```

信号处理工具箱中的函数 kaiserord 也可用凯泽公式来估计滤波器的阶数。它可按如下形式之一使用:

```
[N Wn,beta,ftype] = kaiserord(fedge,aval,dev)
[N Wn,beta,ftype] = kaiserord(fedge,aval,dev,FT)
c = kaiserord(fedge,aval,dev,FT,'cell')
```

其中 FT 是单位为 Hz 的抽样频率,若未被指定,则其默认值是 2 Hz;fedge 是单位为 Hz 的频带边界频率向量,在 0 和 FT/2 之间递增;而 aval 是一个向量,用于指定 fedge 所给出的频带

边界处的幅度响应的期望值。fedge 的长度是 aval 的长度的两倍减 2,因此必须是偶数。dev 是每个频带内允许的最大偏差或波纹向量,单位为 dB。若所指定的偏差不相等,则对所有频带使用最小的一个。输出数据以需要的格式存储在 fir1 中,其中归一化频带边界 Wn 和参数 beta 用于计算式(7.36)所给出的窗系数。字符串 ftype 为 fir1 指定滤波器类型。对于高通滤波器设计,它是 high,而对于带阻滤波器设计,它是 stop。kaiserord 的最后一种形式指定一个单元矩阵,它是 fir1 的参数。

MATLAB 函数 remeaord 可实现式(7.8)给出的公式。它可使用如下形式中的一种:

```
[N fts,mval,wgts] = remezord(fedge,aval,dev)
[N fts,mval,wgts] = remezord(fedge,aval,dev,FT)
```

其中 FT 是单位为 Hz 的抽样频率,若未被指定,它的默认值是 2 Hz;fedge 是单位为 Hz 的频带边界频率向量,在 0 和 FT/2 之间递增;aval 是一个向定,用于指定 fedge 给出的频带边界处的幅度响应的期望值。fedge 的长度是 aval 的长度的两倍减 2,因此必须是偶数。Dev 是每个频带内允许的最大偏差或波纹向量,单位为 dB。remezord 的第三种形式为

```
c = remezord(fedge,aval,dev,FT,'cell')
```

并指定一种单元矩阵,该矩阵的元素是 remez 的参数。

在某些情况下,用任意一种方法确定的阶数 N 可能得不到满足原指标的有限冲激响应滤波器。若确实得不到,阶数应逐渐地增加或减少 1,直到满足指标。而且,对于频带很窄或很宽的有限冲激响应滤波器,阶数估计可能会非常不准确。

例 7.3　吉布斯现象

通过截短由式(7.16)、式(7.18)、式(7.20)、式(7.22)、式(7.24)和式(7.26)给出的理想滤波器的冲激响应,来设计得到有限冲激响应滤波器,然后计算它们的频率响应,可以说明吉布斯现象的发生。低通滤波器的截短的冲激响应系数可在 MATLAB 中使用函数 sinc 产生。该函数通过简单的修改,也可用于产生一种高通、带通或带阻滤波器的截短的冲激响应系数。

习题

Q7.9　使用函数 sinc 编写一个 MATLAB 程序,以产生截止频率在 $\omega_c = 0.4\pi$ 处、长度分别为 81,61,41 和 21 的四个零相位低通滤波器的冲激响应系数,然后计算并画出它们的幅度响应。使用冒号":"运算符从长度为 81 的滤波器的冲激响应系数中抽出较短长度滤波器的冲激响应系数。在每一个滤波器的截止频率两边研究频率响应的摆动行为。波纹的数量与滤波器的长度之间有什么关系?最大波纹的高度与滤波器的长度之间有什么关系?你将怎样修改上述程序以产生一个偶数长度的零相位低通滤波器的冲激响应系数?

Q7.10　使用函数 sinc 编写一个 MATLAB 程序,以产生一个截止频率在 $\omega_c = 0.4\pi$ 处、长度为 45 的零相位高通滤波器的冲激响应系数,计算并画出其幅度响应。在每一个滤波器的截止频率两边研究频率响应的摆动行为。你将怎样修改上述程序以产生一个偶数长度的零相位高通滤波器的冲激响应系数?

Q7.11 编写一个 MATLAB 程序,以产生长度分别为 81,61,41 和 21 的四个零相位微分器的冲激响应系数,计算并画出它们的幅度响应。下面的代码段显示了怎样产生一个长度为 2M+1 的微分器。

```
n=1:M;
b=cos(pi*n)./n;
num=[-fliplr(b) 0 b];
```

对于每种情况,研究微分器的频率响应的摆动行为。波纹的数量与微分器的长度之间有什么关系?最大波纹的高度与滤波器的长度之间有什么关系?

Q7.12 编写一个 MATLAB 程序,以产生长度分别为 81,61,41 和 21 的四个离散时间希尔伯特变换器的冲激响应系数,计算并画出它们的幅度响应。下面的代码段显示了怎样产生一个长度为 2M+1 的希尔伯特变换器。

```
n=1:M;
c=sin(pi*n)./2;
b=2*(c.*c)./(pi*n);
num=[-fliplr(b) 0 b];
```

对于每种情况,研究希尔伯特变换器的频率响应的摆动行为。波纹的数量与希尔伯特变换器的长度之间有什么关系?最大波纹的高度与滤波器的长度之间有什么关系?

例 7.4 有限冲激响应滤波器的阶数估计

习题

Q7.13 使用函数 kaiord,估计具有以下指标的线性相位低通有限冲激响应滤波器的阶数:通带边界为 2 kHz,阻带边界为 2.5 kHz,通带波纹 $\delta_p=0.005$,阻带波纹 $\delta_s=0.005$,抽样率为 10 kHz。在函数 kaiord 中,命令 ceil 和 nargin 的作用是什么?

Q7.14 对下面的情况重做习题 Q7.13:(a)20 kHz 的抽样率,(b) $\delta_p=0.002$ 和 $\delta_s=0.002$,(c)阻带边界为 2.3 kHz。把每一种情况中得到的滤波器长度与习题 Q7.13 中得到的相比较。评论抽样率、波纹以及过渡带宽对滤波器阶数的影响。

Q7.15 用函数 kaiserord 重做习题 Q7.13。把所得到的滤波器阶数值与习题 Q7.13 中得到的相比较。

Q7.16 用函数 remezord 重做习题 Q7.13。把所得到的滤波器阶数值与习题 Q7.13 和习题 Q7.15 中得到的相比较。

Q7.17 使用函数 kaiord,估计具有以下指标的线性相位带通有限冲激响应滤波器的阶数:通带边界为 1.8 kHz 和 3.6 kHz,阻带边界为 1.2 kHz 和 4.2kHz,通带波纹 $\delta_p=0.1$,阻带波纹 $\delta_s=0.02$,抽样率为 12 kHz。

Q7.18 用函数 kaiserord 重做习题 Q7.17。把所得到的滤波器的阶数值与习题 Q7.17 中得到的相比较。

Q7.19 用函数 remezord 重做习题 Q7.17。把所得到的滤波器的阶数值与习题 Q7.17 和习题 Q7.18 中得到的相比较。

例 7.5　有限冲激响应滤波器设计

习题

Q7.20　使用函数 fir1,设计一个线性相位有限冲激响应低通滤波器,使其满足习题 Q7.13 给出的指标,并画出其增益和相位响应。使用习题 Q7.13 中用凯泽公式估计出的阶数。用表格形式显示滤波器系数。你的设计满足指标吗?若不满足,调整滤波器阶数直到设计满足指标。满足指标的滤波器阶数是多少?

Q7.21　用下面的每一种窗函数重做习题 Q7.20:汉宁窗、布莱克曼窗和多尔夫-切比雪夫窗。

Q7.22　用函数 remez 重做习题 Q7.20。

Q7.23　用凯泽窗设计一个有限冲激响应低通滤波器。滤波器的指标是:$\omega_p = 0.31$,$\omega_s = 0.41$,$A_s = 50$ dB。注意,函数 kaiser 需要参数 β 及阶数 N 的值,它们必须先用式(7.36)和式(7.37)分别算出。你的设计满足指标吗?

Q7.24　用函数 kaiserord 和 fir1 重做习题 Q7.23。

Q7.25　用 fir2 设计一个 95 阶有限冲激响应滤波器,它具有三个不同的常数幅度级:在频率范围 0 到 0.25 中为 0.4,在频率范围 0.3 到 0.45 中为 1.0,在频率范围 0.5 到 1.0 中为 0.8。画出所设计的滤波器的幅度响应。你的设计满足指标吗?

Q7.26　使用 remez 设计有限冲激响应带通滤波器,使其具有在习题 Q7.17 中给出的指标以及使用函数 kaiserord 得到的阶数。在函数 remez 中需要的向量 wgts 为

wgts = $\max(\delta_p, \delta_s)[1/\delta_p, 1/\delta_s]$

你的设计满足指标吗?若不满足,调整滤波器阶数直到设计满足指标。满足指标的滤波器阶数是多少?

Q7.27　用 remez 设计具有如下指标的有限冲激响应带通滤波器:通带边界为 1.8 kHz 和 3.0 kHz,阻带边界为 1.5 kHz 和 4.2 kHz,通带波纹 $\delta_p = 0.1$,阻带波纹 $\delta_s = 0.02$,抽样率为 12 kHz。用 kaiserord 估计滤波器的阶数。你的设计是一个最优有限冲激响应滤波器吗?你的设计满足指标吗?若不满足,增加滤波器阶数在满足指标方面有用吗?指标由一个较低阶数的滤波器来满足而不是由 kaiserord 得到的来满足吗?在不等过渡带的情形下,用 remez 设计的滤波器可能在较大的过渡带宽中以增益响应表现不满意的行为。改进行为的一种方法是:通过移动阻带边界减少过渡带宽,直到使设计在过渡带中以平滑的下降来满足指标。在通带边界保持固定的情况下,尝试这种方法并确定新的指标,它在过渡带中提供平滑的下降。

7.6　基础知识阅读

[1] A. Antoniou. *Digital Filters: Analysis, Design, and Applications.* McGraw-Hill, New York NY, second edition, 1993. Chs. 7, 9.

[2] E. Cunningham. *Digital Filtering: An Introduction.* Houghton-Mifflin, Boston MA, 1992. Secs. 4.2, 4.3, 4.5, 4.6, 5.2–5.5.

[3] D.J. DeFatta, J.G. Lucas, and W.S. Hodgkiss. *Digital Signal Processing: A System Design Approach.* Wiley, New York NY, 1988. Chs. 4, 5.

[4] L.B. Jackson. *Digital Filters and Signal Processing.* Kluwer, Boston, MA, third edition, 1996. Chs. 8, 9.

[5] R. Kuc. *Introduction to Digital Signal Processing.* McGraw-Hill, New York NY, 1988. Chs. 8, 9.

[6] L.C. Ludeman. *Fundamentals of Digital Signal Processing.* Harper & Row, New York NY. 1986. Ch. 4.

[7] S.K. Mitra. *Digital Signal Processing: A Computer-Based Approach.* McGraw-Hill, New York NY, 1998. Ch. 7.

[8] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing.* Prentice-Hall, Englewood Cliffs NJ, 1989. Ch. 7.

[9] S.J. Orfanidis. *Introduction to Signal Processing.* Prentice-Hall, Englewood Cliffs NJ, 1996. Ch. 2 and Chs. 10, 11.

[10] B. Porat. *A Course in Digital Signal Procesing.* Wiley, New York NY, 1996. Secs. 9.2–9.6, 10.9.

[11] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs NJ, third edition, 1996, Ch. 8.

[12] R.A. Roberts and C.T. Mullis. *Digital Signal Processing.* Addison-Wesley, Reading MA, 1987. Ch. 6.

第 8 章　数字滤波器实现

8.1　导言

在以适当的滤波器形式实现了通过逼近给定的频率响应指标而得到的数字传输函数 $G(z)$后，我们接下来在硬件或软件形式中实现该结构。在本次实验练习中，我们将使用 MATLAB 来研究无限冲激响应或有限冲激响应数字滤波器结构的软件实现。

8.2　基础知识回顾

R8.1　验证结构的一种简单技术基于传输函数系数与滤波器冲激响应样本之间的卷积和关系[Mit77b]。假设 $H(z)$是一个 N 阶无限冲激响应传输函数

$$H(z)=\frac{p_0+p_1\,z^{-1}+p_2\,z^{-2}+\ldots+p_N\,z^{-N}}{1+d_1\,z^{-1}+d_2\,z^{-2}+\ldots+d_N\,z^{-N}} \tag{8.1}$$

若$\{h[n]\}$表示其单位样本响应，则有

$$H(z)=\sum_{n=0}^{\infty}h[n]\,z^{-n} \tag{8.2}$$

由式(8.1)及式(8.2)可以推出

$$p_n=h[n]\circledast d_n,\qquad n=0,1,2,\cdots \tag{8.3}$$

式(8.3)中前 $2N+1$ 个等式可用矩阵形式表示为

$$\mathbf{p}=\mathbf{H}_1\begin{bmatrix}1\\ \mathbf{d}\end{bmatrix} \tag{8.4}$$

$$\mathbf{0}=[\mathbf{h}\quad \mathbf{H}_2]\begin{bmatrix}1\\ \mathbf{d}\end{bmatrix} \tag{8.5}$$

其中，

$$\mathbf{p}=\begin{bmatrix}p_0\\ p_1\\ \vdots\\ p_N\end{bmatrix},\quad \mathbf{d}=\begin{bmatrix}d_1\\ d_2\\ \vdots\\ d_N\end{bmatrix},\quad \mathbf{0}=\begin{bmatrix}0\\ \vdots\\ 0\end{bmatrix} \tag{8.6}$$

$$\mathbf{H}_1=\begin{bmatrix}h[0] & 0 & \ldots & 0\\ h[1] & h[0] & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots\\ h[N] & h[N-1] & \ldots & h[0]\end{bmatrix} \tag{8.7}$$

$$\mathbf{h}=\begin{bmatrix}h[N+1]\\ \vdots\\ h[2N]\end{bmatrix},\qquad \mathbf{H}_2=\begin{bmatrix}h[N] & \ldots & h[1]\\ \vdots & \ddots & \vdots\\ h[2N-1] & \ldots & h[N]\end{bmatrix} \tag{8.8}$$

求解式(8.5)首先得到由分母系数组成的向量 $\mathbf{d}$

$$\mathbf{d} = -\mathbf{H}_2^{-1}\mathbf{h} \tag{8.9}$$

将式(8.9)代入式(8.4)中来确定包含分子系数的向量 $\mathbf{p}$

$$\mathbf{p} = \mathbf{H}_1 \begin{bmatrix} 1 \\ -\mathbf{H}_2^{-1}\mathbf{h} \end{bmatrix} \tag{8.10}$$

R8.2 可调数字滤波器的设计基础是复变量 z 的谱变换,它通过将 z 替换为 $F(z)$,可将具有给定截止频率的数字滤波器的实现调整为具有不同截止频率的另一个实现。因此,若 $G_{old}(z)$是原实现的传输函数,新结构的传输函数是 $G_{new}(z)$,则有

$$G_{new}(z) = G_{old}(z)\big|_{z^{-1}=F^{-1}(z)} \tag{8.11}$$

其中 $F^{-1}(z)$是一个被适当选择的稳定全通函数,其系数是调节参数[Con70]。实现这种转换的一种直接方法是,在 $G_{old}(z)$的实现中用实现 $F^{-1}(z)$的一个全通结构代替每一个延时块。然而,这种方法通常会导致实现 $G_{new}(z)$时具有不能被实现的无延时环结构。

应用到并联全通实现的上述方法的一种非常简单的修改,不会导致具有无延时环的结构[Mit90]。形式

$$G(z) = \frac{1}{2}\{A_0(z) + A_1(z)\} \tag{8.12}$$

可以实现具有对称分子的有界实传输函数 $G(z)$,以及具有反对称分子的功率互补传输函数$H(z)$。其中 $A_0(z)$和 $A_1(z)$是稳定的全通滤波器。上面关于 $G(z)$的条件同样适用于奇数阶低通巴特沃兹、切比雪夫和椭圆传输函数。

全通滤波器 $A_0(z)$和 $A_1(z)$以形如图 6.9~图 6.11 中描述的一阶和二阶部分的级联实现。这些结构对一阶全通函数的实现仅用一个乘法器和一个延时,对二阶全通函数的实现仅用两个乘法器和两个延时。

对于通过并联全通结构实现的低通无限冲激响应滤波器的截止频率的调整,低通到低通变换由下式给出[Con70]:

$$z^{-1} \to F(z^{-1}) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \tag{8.13}$$

其中参数 α 通过

$$\alpha = \frac{\sin[(\omega_c - \hat{\omega}_c)/2]}{\sin[(\omega_c + \hat{\omega}_c)/2]} \tag{8.14}$$

分别与旧的截止频率 ω_c 和新截止频率$\hat{\omega}_c$ 相关。

在 1 型一阶全通传输函数[①]

$$a_1(z) = \frac{d_1 + z^{-1}}{1 + d_1 z^{-1}} \tag{8.15}$$

① 见 R6.13。

中代入变换式(8.13),可得变换后的一阶全通传输函数为

$$\hat{a}_1(z) = a_1(z)\big|_{z^{-1}=\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}} \approx \frac{[d_1+\alpha(d_1^2-1)]+z^{-1}}{1+[d_1+\alpha(d_1^2-1)]\,z^{-1}} \tag{8.16}$$

假设 α 较小。$\hat{a}_1(z)$是系数为 α 的线性函数的一个 1 型一阶全通传输函数。在 3 型二阶全通传输函数[①]

$$a_2(z) = \frac{d_2+d_1z^{-1}+z^{-2}}{1+d_1z^{-1}+d_2z^{-2}} \tag{8.17}$$

中,式(8.13)的变换得出

$$\begin{aligned}\hat{a}_2(z) = a_2(z)\big|_{z^{-1}=\frac{z^{-1}-\alpha}{1-\alpha z^{-1}}} &= \frac{d_2+d_1(\frac{z^{-1}-\alpha}{1-\alpha z^{-1}})+(\frac{z^{-1}-\alpha}{1-\alpha z^{-1}})^2}{1+d_1(\frac{z^{-1}-\alpha}{1-\alpha z^{-1}})+d_2(\frac{z^{-1}-\alpha}{1-\alpha z^{-1}})^2}\\ &\approx \frac{[d_2+\alpha d_1(d_2-1)]+[d_1-2\alpha(1+d_2)+\alpha d_1^2]\,z^{-1}+z^{-2}}{1+[d_1-2\alpha(1+d_2)+\alpha d_1^2]\,z^{-1}+[d_2+\alpha d_1(d_2-1)]\,z^{-2}}\end{aligned} \tag{8.18}$$

可以看出它是系数为 α 的线性函数的一个 3 型二阶全通传输函数。

通过对可调低通无限冲激响应滤波器应用一个低通到带通转换[Con70]:

$$z^{-1} \to F(z^{-1}) = -z^{-1}\,\frac{z^{-1}+\beta}{1+\beta z^{-1}} \tag{8.19}$$

我们可设计一个可调带通滤波器,其中心频率 ω_o 可通过调整参数 $\beta=\cos\omega_o$ 来调整,而其带宽通过改变 α 来调整。与式(8.13)所示的低通到低通转换不同,通过将每个延时替换为一个全通结构级联一个延时,该变换可以在实现可调低通滤波器的结构上直接实现。

R8.3 设计可调有限冲激响应滤波器的一种非常简单的方法是基于加窗的傅里叶级数方法[Jar88]。对于具有零相位响应

$$H_d(e^{j\omega}) = \begin{cases} 1, & 0\leqslant\omega\leqslant\omega_c \\ 0, & \omega_c\leqslant\omega\leqslant\pi \end{cases} \tag{8.20}$$

的理想有限冲激响应低通滤波器,冲激响应系数为

$$h_d[n] = \frac{\sin(\omega_c n)}{\pi n},\quad 0\leqslant|n|<\infty \tag{8.21}$$

截短上式可得到一个可实现的逼近系数

$$h_{LP}[n] = \begin{cases} c[n]\,\omega_c, & n=0 \\ c[n]\,\sin(\omega_c n), & 1\leqslant|n|\leqslant N \\ 0, & \text{其他} \end{cases} \tag{8.22}$$

其中 ω_c 是 6 dB 截止频率且

$$c[n] = \begin{cases} 1/\pi, & \text{for } n=0, \\ 1/\pi n, & \text{for } 1\leqslant|n|\leqslant N \end{cases} \tag{8.23}$$

① 见 R6.15。

从上面的内容可以推出,一旦针对给定的截止频率设计出一个有限冲激响应低通滤波器,它可通过改变 ω_c 并按照上述表达式重算滤波器系数来简单地调整。可以看出,通过用任何有限冲激响应滤波器设计方法生成具有式(8.22)所示原型滤波器的系数的方程并求解 $c[n]$,则式(8.22)也可用于设计可调有限冲激响应低通滤波器。因此,若 $h_{LP}[n]$表示对截止频率 ω_c 设计的原型低通滤波器的系数,由式(8.22)可知常数 $c[n]$为

$$c[0]=\frac{h_{LP}[0]}{\omega_c},\quad c[n]=\frac{h_{LP}[n]}{\sin(\omega_c n)},\quad 1\leqslant |n|\leqslant N \tag{8.24}$$

则具有截止频率 $\hat{h}_{LP}[n]$的转换后的有限冲激响应滤波器的系数 $\hat{\omega}_c$ 由下式给出

$$\hat{h}_{LP}[0]=c[0]\hat{\omega}_c=\left(\frac{\hat{\omega}_c}{\omega_c}\right)h_{LP}[0],\quad \hat{h}_{LP}[n]=c[n]\sin(\hat{\omega}_c n)=\left(\frac{\sin(\hat{\omega}_c n)}{\sin(\omega_c n)}\right)h_{LP}[n],\quad 1\leqslant|n|\leqslant N \tag{8.25}$$

该调整过程对用具有相等通带和阻带波纹的滤波器工作良好。原型滤波器的设计应该不使其系数值与零太接近。

R8.4 长度为 N 的序列 $x[n]$,$0\leqslant n\leqslant N-1$ 的 N 点离散傅里叶变换 $X[k]$,$0\leqslant k\leqslant N-1$,由其 z 变换 $X[z]$在单位圆上 N 个等间隔点 $z=e^{j2\pi k/N}$处求出的 N 个样本给出,$0\leqslant k\leqslant N-1$:

$$X[k]=X(z)|_{z=e^{j2\pi k/N}}=\sum_{n=0}^{N-1}x[n]e^{-j2\pi kn/N},\quad 0\leqslant k\leqslant N-1 \tag{8.26}$$

对长度为 N 的复数值序列,使用式(8.26)的 N 点离散傅里叶变换的一个直接计算,需要 N^2 个复数乘法和 $N(N-1)\approx N^2$ 个复数加法。

若 N 是偶数,$X(z)$可以表示为两频带多相形式(见 R10.9):

$$X(z)=X_0(z^2)+z^{-1}X_1(z^2) \tag{8.27}$$

其中 $X_0(z)$是由偶序号样本形成的长度为$(N/2)$的子序列的 z 变换,而 $X_1(z)$是由 $x[n]$ 的奇序号样本形成的长度为$(N/2)$的子序列的 z 变换,即

$$X_0(z)=\sum_{n=0}^{\frac{N}{2}-1}x_0[n]z^{-n}=\sum_{n=0}^{\frac{N}{2}-1}x[2n]z^{-n} \tag{8.28}$$

$$X_1(z)=\sum_{n=0}^{\frac{N}{2}-1}x_1[n]z^{-n}=\sum_{n=0}^{\frac{N}{2}-1}x[2n+1]z^{-n} \tag{8.29}$$

因此,N 点离散傅里叶变换 $X[k]$可用下式计算:

$$X[k]=X_0[\langle k\rangle_{N/2}]+W_N^k X_1[\langle k\rangle_{N/2}],\quad 0\leqslant k\leqslant N-1 \tag{8.30}$$

其中 $X_0[k]$和 $X_1[k]$分别是长为$\frac{N}{2}$的序列 $x_0[n]$和 $x_1[n]$的$\frac{N}{2}$点离散傅里叶变换,。使用式(8.30)的 N 点离散傅里叶变换的一个实现,现在需要 $N+\frac{N^2}{2}$ 个复数乘法和大约 $N+\frac{N^2}{2}$个复数加法。若 N 是2的幂,则该过程可继续,直到 N 点离散傅里叶变换计算

简化为长度为 2 的幂的离散傅里叶变换的加权和。这是按时间抽取快速傅里叶变换算法的基本思想。可以看到在通常情况中,复数乘法和复数加法的总数量是 $N(\log_2 N)$。通过使用 W_N 的对称性质,可进一步简化计算的复杂性。

R8.5 数 x 的正弦可用幂级数逼近[Abr72]为

$$\sin(x) \approx x - 0.166\,667x^3 + 0.008\,333x^5 - 0.000\,198\,4x^7 + 0.000\,002\,7x^9 \quad (8.31)$$

其中自变量 x 的单位是弧度,而其范围被限制在第一象限,即从 0 到 $\pi/2$。若 x 在该范围之外,则其正弦可使用恒等式 $\sin(-x) = -\sin(x)$ 及 $\sin(\frac{\pi}{2}+x) = \sin(\frac{\pi}{2}-x)$来计算。

R8.6 数 x, $-1 \leqslant x \leqslant 1$ 的反正切可用下式计算[Abr72]:

$$\begin{aligned}\arctan(x) \approx {} & 0.999\,866x - 0.330\,299\,5x^3 + 0.180\,141x^5 \\ & - 0.085\,133x^7 + 0.020\,835\,1x^9\end{aligned} \quad (8.32)$$

若 $1 \leqslant x$ 或 $x \leqslant -1$,可用恒等式 $\arctan(x) = -\arctan(1/x)$计算 x 的反正切。

R8.7 位于区间 $0.5 \leqslant x \leqslant 1$ 中的正数 x 的平方根,可用截短的多项式近似值求取[Mar92]:

$$\begin{aligned}\sqrt{x} \approx {} & 0.207\,580\,6 + 1.454\,895x - 1.344\,91x^2 \\ & + 1.106\,812x^3 - 0.536\,499x^4 + 0.112\,121\,6x^5\end{aligned} \quad (8.33)$$

若 x 在从 0.5 到 1 的范围之外,则可乘以一个二进制常数 K^2,使乘积 $x' = K^2x$ 在所需的范围内,用式(8.25)计算$\sqrt{x'}$,然后确定$\sqrt{x} = \sqrt{x'}/K$。

8.3 用到的 MATLAB 命令

这次练习中将使用如下的 MATLAB 命令:

通用命令

disp　length

运算符和特殊符号

:　.　+　-　*　/　;

%　=　<　~=

语言构造和调试

end　for　function　if

基本矩阵和矩阵控制

fliplr　nargin　pi　:　zeros

基本函数

abs　cos　log10　real　sin　sqrt

数据分析

conv　max

二维图形

```
axis     clf    grid    legend    plot
xlabel   ylabel stem    title
```

信号处理工具箱

```
butter    buttord    ellip    ellipord    filter
freqz     remez      tf2zp    zp2sos
```

关于这些命令的详细说明,请参阅 *MATLAB Reference Guide*[Mat94]和 *Signal Processing Toolbox User's Guide*[Mat96],或在命令窗口中键入 help *命令名称*。附录 B 给出了此处使用的 MATLAB 函数的简要解释。

8.4 无限冲激响应滤波器的仿真

N 阶因果无限冲激响应传输函数使用传输函数 $H(z)$:

$$H(z)=\frac{\sum_{k=0}^{N}p_k z^{-k}}{1+\sum_{k=1}^{N}d_k z^{-k}} \tag{8.34}$$

描述。利用不同的方法实现 $H(z)$,可得到不同的结构。这里,我们将仅对少许此类结构用 MATLAB 来说明无限冲激响应滤波器的仿真。MATLAB 信号处理工具箱中的函数 filter,可用于仿真转置的直接 II 型结构的因果无限冲激响应滤波器的实现。一个三阶滤波器转置的直接 II 型结构如图 6.6(b)所示[①]。这个函数的基本形式如下:

```
y = filter(num,den,x)
[y,sf] = filter(num,den,x,si)
```

分子和分母系数分别包含在向量 num 和 den 中。这些向量的大小不必相同。输入向量是 x,而由滤波算法产生的输出向量是 y。若 den 的第一个系数不等于 1,则程序会自动地归一化 num 和 den 中的所有滤波器系数,使之等于 1。

在函数 filter 的第二种形式里,延时(状态)变量的初始条件可通过自变量 si 指定。而且,函数 filter 可通过输出向量 sf 返回延时(状态)变量的终值。初始(最终)条件向量 si (sf)的大小比滤波器系数向量 num 和 den 的最大数小 1。若被处理的输入向量很长,且需要在进程中分割成小数据块进行处理,则以向量 sf 给出的状态变量最终值是有用的。在这种情形下,在处理完输入数据的第 i 块后,最终状态向量 sf 以初始状态向量 si 作为输入数据的第($i+1$)块的处理被送入,依次类推。

为了实现以直接 II 型实现的因果关系无限冲激响应滤波器,可使用下面给出的函数 direct2。

```
function [y,sf] = direct2(p,d,x,si);
% Y = DIRECT2(P,D,X) 以用向量 P 和 D
% 描述的滤波器,对输入数据向量 X
```

---

① 见 R6.9。

```
    %滤波,以产生滤波后的数据 Y。
    %滤波器是差分方程
    %y(n) = p(1) * x(n) + p(2) * x(n-1) + … + p(np+1) * x(n-np)
    %          - d(2) * y(n-1) - … - d(nd+1) * y(n-nd)
    %的"直接 II 型"实现。
    %[Y,SF] = DIRECT2(P,D,X,SI)
    %给出延时的初始和最终条件 SI 和 SF 的接口。
    dlen = length(d); plen = length(p);
    N = max(dlen,plen); M = length(x);
    sf = zeros(1,N-1); y = zeros(1,M);
    if nargin ~= 3,
    sf = si;
    end
    if dlen < plen,
        d = [d zeros(1,plen - dlen)];
        else
        p = [p zeros(1,dlen - plen)];
    end
    p = p/d(1); d = d/d(1);
    for n = 1:M;
        wnew = [1 - d(2:N)] * [x(n) sf]';
        k = [wnew sf];
        y(n) = K * p';
        sf = [wnew sf(1:N-2)];
    end
```

我们可使用程序 P6.1 为生成无限冲激响应传输函数的级联实现。同样,为了生成并联实现,可使用程序 P6.2。在级联或并联形式结构的仿真中,单独的一阶和二阶部分可用直接 II 型(使用函数 direct2)或用转置的直接 II 型(使用函数 filter)实现。

例 8.1　结构仿真和验证

所仿真的结构可在 MATLAB 中用下面给出的函数 strucver 计算其传输函数来验证。

```
function [p,d] = strucver(ir,N);
H = zeros(2 * N + 1,N + 1);
H(:,1) = ir';
for n = 2:N + 1;
    H(:,n) = [zeros(1,n - 1) ir(1:2 * (N + 1) - n)]';
end
H1 = zeros(N + 1,N + 1);
for k = 1:N + 1;
    H1(k,:) = H(k,:);
end
H3 = zeros(N,N + 1);
for k = 1:N;
    H3(k,:) = H(k + N + 1,:);
end
H2 = H3(:,2:N + 1);
hf = H3(:,1);
%计算分母系数
d = - (inv(H2)) * hf;
```

```
%计算分子系数
p = H1 * [1;d];
d = [1; d];
```

程序 P8.1 示例了一个因果无限冲激响应滤波器的设计及其用直接 II 型的仿真。它使用前面描述的函数 strucver 和 direct2。

```
%程序 P8.1
Wp = [0.4 0.5]; Ws = [0.1 0.8]; Rp = 1; Rs = 30;
[N1,Wn1] = buttord(Wp,Ws,Rp,Rs);
[num,den] = butter(N1,Wn1);
disp('分子系数是');disp(num);
disp('分母系数是');disp(den);
impres = direct2(num,den,[1 zeros(1,4 * N1)]);
[p,d] = strucver(impres,2 * N1);
disp('实际分子系数是'); disp(p');
disp('实际分母系数是'); disp(d');
```

习题

Q8.1 程序 P8.1 设计了什么类型的滤波器？其指标是什么？滤波器的阶数是多少？为了验证仿真,需计算多少个冲激响应样本？仿真是正确的吗？

Q8.2 用转置的直接 II 型仿真滤波器,修改程序 P8.1 并运行修改后的程序。仿真是正确的吗？

Q8.3 生成在习题 Q8.1 中产生的传输函数的一个级联实现,并编写出一个程序来仿真它,其中每个单独的部分用一个直接 II 型实现。验证仿真。

Q8.4 以级联部分相反的顺序重做习题 Q8.3。

Q8.5 生成在习题 Q8.1 中产生的传输函数的一个并联 I 型实现,并编写出一个程序来仿真它,其中每个单独的部分用一个直接 II 型实现。验证仿真。

Q8.6 生成在习题 Q8.1 中产生的传输函数的一个并联 II 型实现,并编写出一个程序来仿真它,其中每个单独的部分用一个直接 II 型实现。验证仿真。

编写 MATLAB 程序来仿真任何数字滤波器结构是相当直接的。我们接下来考虑无限冲激响应传输函数的 Gray-Markel 级联格型实现的仿真。图 8.1 显示了因果三阶传输函数

$$H(z)=\frac{0.44\,z^{-1}+0.36\,z^{-2}+0.02\,z^{-3}}{1+0.4\,z^{-1}+0.18\,z^{-2}-0.2\,z^{-3}} \tag{8.35}$$

的级联格型实现。图 8.1 的乘法器系数给出为

$$d_3=-0.2,\quad d_2=0.270\,833,\quad d_1=0.357\,36,$$
$$\alpha_1=0.02,\quad \alpha_2=0.352,\quad \alpha_3=0.276\,533,\quad \alpha_4=-0.190\,16$$

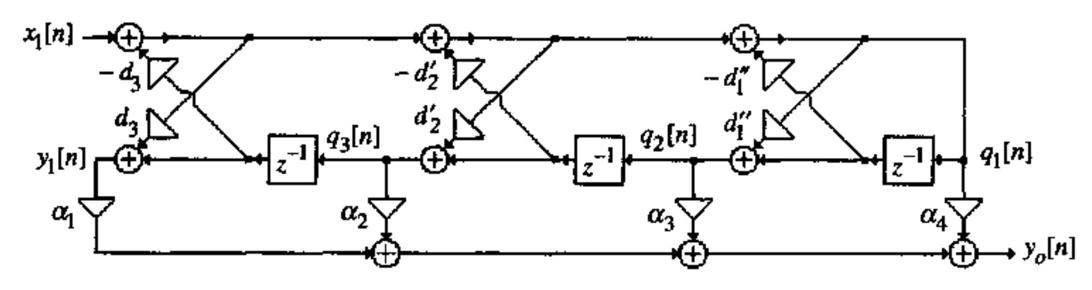


图 8.1 式(8.35)所示 IIR 传输函数的 Gray-Markel 级联格型实现

观察图 8.1,得到描述这个结构的方程为

$$\begin{aligned}
w_3[n] &= x_1[n] - d_3 q_3[n-1] \\
w_2[n] &= w_3[n] - d_2' q_2[n-1] \\
q_1[n] &= w_2[n] - d_1'' q_1[n-1] \\
q_2[n] &= d_1'' q_1[n] + q_1[n-1] \\
q_3[n] &= d_2' w_2[n] + q_2[n-1] \\
y_1[n] &= d_3 w_3[n] + q_3[n-1] \\
y_o[n] &= \alpha_1 y_1[n] + \alpha_2 q_3[n] + \alpha_3 q_2[n] + \alpha_4 q_1[n]
\end{aligned}$$

以下给出该结构仿真的程序 P8.2。输入向量 x1 由一个单位样本序列的前 7 个系数组成,输出向量 y0 由前 7 个冲激响应系数组成,它们然后用于确定用 R8.1 中描述的方法执行的实际传输函数的分子和分母系数。

```
% 程序 P8.2
% 无限冲激响应级联格型结构的仿真
%
x1 = [1 zeros(1,6)]; % 产生单位冲激序列
q3old = 0; q2old = 0; q1old = 0; % 初始条件
% 输入滤波器系数
D1 = 0.357377; D2 = 0.27083; D3 = -0.2;
alpha1 = 0.02; alpha2 = 0.352;
alpha3 = 0.276533; alpha4 = - 0.19016;
% 计算前 7 个冲激响应样本
for n = 1:7
    w3 = x1(n) - D3 * q3old;
    w2 = w3 - D2 * q2old;
    q1new = w2 - D1 * q1old;
    q2new = D1 * q1new + q1old;
    q3new = D2 * w2 + q2old;
    y1 = D3 * w3 + q3old;
    yo(n) = alpha1 * y1 + alpha2 * q3new +
        alpha3 * q2new + alpha4 * q1new;
    q3old = q3new; q2old = q2new; q1old = q1new;
end
[num,den] = strucver(yo,3);
disp('分子系数是');disp(num');
disp('分母系数是');disp(den')
```

习题

Q8.7 运行程序 P8.2 并产生被仿真的级联格型结构的传输函数。该传输函数与式(8.35)中产生的相同吗?

Q8.8 使用 MATLAB,用下面的指标产生一个椭圆低通滤波器的传输函数:通带边界在 0.4π 处,阻带边界在 0.6π 处,通带波纹为 0.5dB,最小阻带衰减为 28 dB。使用程序 P6.3 生成其 Gray-Markel 级联格型实现。用程序 P8.2 仿真并验证该实现。

例 8.2 滤波的说明

程序 P8.3 用于说明因果无限冲激响应滤波器的设计、其转置的直接 II 型的仿真,以及其对一个信号进行滤波的应用。

```
% 程序 P8.3
% 一个无限冲激响应滤波器滤波的说明
%
clf
% 产生输入序列
k = 0:50;
w2 = 0.7 * pi;w1 = 0.2 * pi;
x1 = 1.5 * cos(w1 * k); x2 = 2 * cos(w2 * k);
x = x1 + x2;
% 确定滤波器传输函数
[N,Wn] = ellipord(0.25,0.55,0.5,50);
[num,den] = ellip(N,0.5,50,Wn);
% 产生输出序列
y = filter(num,den,x);
% 画出输入及输出序列
subplot(2,1,1);
stem(k,x); axis([0 50 -4 4]);
xlabel('时间序号 n'); ylabel('振幅')
title('输入序列');
subplot(2,1,2);
stem(k,x); axis([0 50 -4 4]);
xlabel('时间序号 n '); ylabel('振幅')
title('输出序列');
```

习题

Q8.9 程序 P8.3 设计了什么类型的滤波器？其指标是什么？滤波器的阶数是多少？形成输入的正弦序列的频率是多少？

Q8.10 运行程序 P8.3 并产生两个图形。哪种输入成分会在滤波器输出出现？为什么输出序列的开始部分不是一种理想的正弦曲线？修改程序 P8.3，以便只过滤序列 x2[n]。产生的输出序列和预料一样吗？证明你的答案。

无限冲激响应传输函数的级联形式可用函数 zp2sos 从其零极点描述中产生。下面给出的代码段演示了程序 P8.3 中椭圆低通传输函数的每个部分的传输函数的产生，以及其整个传输函数作为一个部分的级联形式实现，其中所有的实现采用级联 II 型。

```
[N,Wn] = ellipord(0.25,0.55,0.5,50);
[num,den] = ellip(N,0.5,50,Wn);
[z,p,const] = tf2zp(num,den);
sos = zp2sos(z,p,const);
row1 = real(sos(1,:));row2 = real(sos(2,:));
num1 = row1(1:3);den1 = row1(4:6);
num2 = row2(1:3);den2 = row2(4:6);
y = direct2(num,den,x);
y1 = direct2(num1,den1,x);y2 = direct2(num2,den2,y1);
```

习题

Q8.11 使用上面的代码段来修改程序 P8.3，以对级联结构所产生的序列滤波，将其输出与单个高阶部分滤波时产生的输出相比较。两个输出之间有不同吗？通过对每一个仿真的传输函数写下表达式，精确地显示所仿真的两个滤波器。

Q8.12 使用函数 strucver 修改在习题 Q8.11 中生成的程序,以验证所仿真的结构。运行修改后的程序。你的仿真是正确的吗?

长输入序列可用重叠相加法滤波,其中输入序列被分割成一组相邻的短输入块,每一块接下来被分别滤波,且输出块中的重叠被适当地相加,以产生长输出序列。这种滤波方法可在 MATLAB 中用函数 filter 的第二种形式实现。这里,在滤波的任何阶段,状态变量向量 sf 的终值在接下来的滤波阶段反馈为初始情况向量 si。

习题

Q8.13 修改程序 P8.3,将输入序列作为一组每个长度为 5 的相邻块进行滤波。运行修改后的程序,并将你的结果与按一个部分滤波输入所产生的结果做比较。

8.5　有限冲激响应数字滤波器的仿真

如本例所示,函数 direct2 和 filter 也可用来实现有限冲激响应数字滤波器。

例 8.3　结构仿真和验证

程序 P8.4 用于说明一个因果有限冲激响应滤波器的设计,以及其在转置的直接 II 型中的仿真。

```
% 程序 P8.4
num = remez(9,[0 0.3 0.5 1],[1 1 0 0]);
disp('滤波器系数是');disp(num);
impres = filter(num,1 ,[1 zeros(1,9)]);
disp('实际滤波器系数是'); disp(impres);
```

习题

Q8.14 程序 P8.4 设计了什么类型的滤波器? 其指标是什么? 滤波器的阶数是多少? 为了验证仿真,需计算多少个冲激响应样本? 仿真是正确的吗?

Q8.15 修改程序 P8.4,用直接 II 型仿真该滤波器,运行后修改的程序。仿真是正确的吗?

Q8.16 使用程序 P8.4 生成在习题 Q8.14 中产生的传输函数的一个级联实现,写出仿真其的一个程序。其中每一个部分以直接 II 型实现。验证仿真。

Q8.7 以级联部分相反的顺序,重做习题 Q8.16。

例 8.4　滤波的说明

习题

Q8.18 修改程序 P8.3,用函数 direct2 对程序 P8.4 中的有限冲激响应滤波器采取直接型实现仿真,并说明其滤波性质。

8.6　可调数字滤波器的设计

在接下来的两个例中,我们将研究可调无限冲激响应和有限冲激响应数字滤波器的设计。

例 8.5　可调无限冲激响应滤波器的设计

程序 P8.5 示例了基于并联全通实现的可调因果无限冲激响应低通滤波器的设计(见 R6.19)。

```
% 程序 P8.5
% 可调无限冲激响应滤波器设计的说明
clf;
[z,p,k] = ellip(5,0.5,40,0.4);
a = conv([1 -p(1)],[1 -p(2)]); b = [1 -p(5)];
c = conv([1 -p(3)],[1 -p(4)]);
w = 0:pi/255:pi;
alpha = [0 0.1 -0.25];
for i = 1:3
    an1 = a(2) + (a(2)*a(2) - 2*(1 + a(3)))*alpha(i);
    an2 = a(3) + (a(3) - 1)*a(2)*alpha(i);
    g = b(2) - (1 - b(2)*b(2))*alpha(i);
    cn1 = c(2) + (c(2)*c(2) - 2*(1 + c(3)))*alpha(i);
    cn2 = c(3) + (c(3) - 1)*c(2)*alpha(i);
    a = [1 an1 an2];b = [1 g]; c = [1 cn1 cn2];
    h1 = freqz(fliplr(a),a,w); h2 = freqz(fliplr(b),b,w);
    h3 = freqz(fliplr(c),c,w);
    ma(i,:) = 20*log10(abs(0.5*(h1.*h2 + h3)));
end
plot(w/pi,ma(1,:),'r-',w/pi,ma(2,:),'b--',w/pi,ma(3,:),'g-.');
grid;
axis([0 1 -80 5]);
xlabel('\omega/\pi');ylabel('增益,dB');
legend('\alpha = 0 ','\alpha = 0.1','\alpha = -0.25');
```

习题

Q8.19 程序 P8.5 设计了什么类型的滤波器作为标定滤波器？其指标是什么？滤波器的阶数是多少？

Q8.20 编写一个 MATLAB 程序来设计这种标定滤波器，并用命令 `zplane` 显示其极点位置。确定两个全通部分的传输函数。

Q8.21 运行程序 P8.5 并显示所有增益响应。所设计的滤波器的截止频率是多少？

Q8.22 修改程序 P8.5 来确定并绘制程序 P8.5 中所设计的每一个滤波器的功率互补滤波器的增益响应。

例 8.6　可调有限冲激响应滤波器的设计

程序 P8.6 示例了基于 R8.3 所列方法的可调因果有限冲激响应低通滤波器的设计。

```
% 程序 P8.6
% 可调有限冲激响应滤波器设计的说明
clf;
w = 0:pi/255:pi;
f = [0 0.36 0.46 1];m = [1 1 0 0];
```

```
b1 = remez(50,f,m);
h1 = freqz(b1,1,w);
m1 = 20 * log10(abs(h1));
n = -25:-1;
c = b1(1:25)./sin(0.41 * pi * n);
wc = [0.31 * pi 0.51 * pi];
for i = 1:2
    d = c. * sin(wc(i) * n);
    q = (b1(26) * wc(i))/(0.4 * pi);
    b2 = [d q fliplr(d)];
    mag(i,:) = 20 * log10(abs(freqz(b2,1,w)));
end
plot(w/pi,mag(1,:),'b-',w/pi,m1,'r-',w/pi,mag(2,:),'g-.');
grid;
axis([0 1 -80 5]);
xlabel('\omega/\pi');ylabel('增益,dB');
legend('\omega_{c} = 0.31 \pi','\omega_{c} = 0.41 \pi','\omega_{c} = 0.51 \pi')
```

习题

Q8.23 程序 P8.6 设计了什么类型的滤波器作为标定滤波器？其指标是什么？滤波器的阶数是多少？

Q8.24 运行程序 P8.6 并显示所有增益响应。所设计的滤波器的截止频率是多少？

Q8.25 修改程序 P8.6 来确定并绘制程序 P8.6 所设计的每一个滤波器的延时互补滤波器的增益响应。

8.7 离散傅里叶变换计算

例 8.7 算法的计算复杂度

习题

Q8.26 编写一个 MATLAB 程序，直接用式(8.26)和用 M 函数 `fft` 来分别计算并显示一个长度为 N 的序列的 N 点离散傅里叶变换。你的程序应该使用命令 `flops` 来确定两种情况下离散傅里叶变换计算中涉及的浮点运算的总数量。输入数据是以向量形式输入的序列，通过它确定序列的长度 N。使用这个程序，分别计算长度为 64，128 和 256 的序列的离散傅里叶变换。评论你的结果。

8.8 函数逼近

例 8.8 三角及其他函数的计算

习题

Q8.27 编写一个 MATLAB 程序，用式(8.31)逼近并画出 x 的函数 $\sin(x)$，其中 x 的单位是弧度，并在范围 $0 \leqslant x \leqslant \pi/2$ 中。同样画出由于逼近产生的误差。运行该程序并产生两个图形。评论你的结果。

Q8.28 编写一个 MATLAB 程序,用式(8.32)逼近并画出 x 的函数 arctan(x),其中 x 在范围 $0 \leqslant x \leqslant 1$ 中。同样画出由于逼近产生的误差。运行该程序并产生两个图形。评论你的结果。

Q8.29 编写一个 MATLAB 程序,用式(8.33)逼近 x 的$\sqrt{x}$函数,画出由于计算$\sqrt{x}$产生的误差,其中 x 在范围 $0.5 \leqslant x \leqslant 1$ 中。运行该程序并产生两个图形。评论你的结果。

8.9 基础知识阅读

[1] S.K. Mitra. *Digital Signal Processing: A Computer-Based Approach.* McGraw-Hill, New York NY, 1998. Secs. 8.1, 8.2, 8.7, 8.8.

第9章　有限字长效应分析

9.1　导言

无论是计算机还是专用硬件，它们的数据都存储在有限字长的存储单元中。因此，无论用计算机软件还是专用硬件来实现，线性时不变离散时间系统的参数和变量只能在一定的范围内取不同的离散值。对于信号变量和滤波器常量，只要量化单位足够小，就可以应用一种基于统计模型的近似理论将连续的信号离散化，这样，得到的结果也可以通过实验仿真来验证。滤波器的结构是由传输函数决定的，要实现滤波器就必须对滤波器系数加以量化，滤波器系数的量化就是引起量化误差的原因之一。在数字系统中处理连续时间信号时，必须先对连续信号进行模数转换，模数转换是引起误差的第二个原因。数字运算过程中的量化问题是引起误差的第三个原因。数字滤波器中另一种类型的误差，是由数字运算量化的非线性引起的。这种误差会在没有输入或者输入信号为常数或正弦信号的情况下，引起滤波器的输出即极限环振荡。事实上，在衡量滤波器的性能时，对不同量化效果的分析，是根据滤波器中数据是定点格式还是浮点格式、负数是如何表示的、采用的量化方法和数字滤波器的结构等因素来综合考虑的。在本章的练习中，我们将学习到上面所提到的量化误差所带来的影响，同时还将研究不同滤波器结构的量化灵敏度问题，其中的量化误差是在定点算法实现的数字滤波器中得到的。

9.2　基础知识回顾

R9.1　为了表示一个有符号的 b 位小数，寄存器的最高位用来表示数字的符号，这一位称为*符号位*。正数的符号位为0，而负数的符号位为1。

R9.2　在一个有限字长的寄存器中，表示一个定点数的量化方法有两种：*截尾*和*舍入*。定点数从$(\beta+1)$位到$(b+1)$位的截尾就是简单地丢弃最低的$(\beta-b)$位。而舍入则是将数字量化到最接近的量化级，一个恰好在两个量化级正中间的数字被舍入到最接近的较高级。因此，若 $a_{-(b+1)}$位是0，舍入等效于截尾，若这位数是1，则应该对截尾后的数字的最低位加1。

R9.3　通过硬件或软件方式，用量化系数实现的数字滤波器的传输函数 $\hat{H}(z)$和量化前的传输函数 $H(z)$是不同的。系数量化会对极点和零点的位置带来影响。实际频率响应 $\hat{H}(e^{j\omega})$会不同于量化前的频率响应 $H(e^{j\omega})$。而且，极点可能会移出单位圆之外，使得数字滤波器成为非稳定系统，而具有非量化系数的原传输函数却是稳定的。

R9.4　由于模数转换器的输入输出特性是非线性的，并且在大多数实际情况中，模拟输入信号预先并不知道，为了分析，假定量化误差 $e[n]$是一个随机信号并且使用如图9.1所示的量化器运算统计模型。

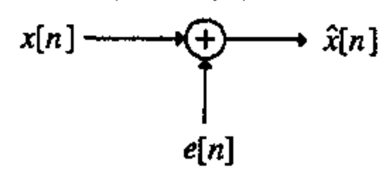


图9.1　模数量化器的统计模型

另外，为了简化分析，做出如下假定：

1. 误差序列$\{e[n]\}$是宽平稳(WSS)白噪声随机过程的一个样本序列,其中每一个样本$e[n]$,如图 9.2 所示,在量化误差范围内服从均匀分布,这里 δ 是量化阶。
2. 误差序列与其相应的输入序列$\{x[n]\}$不相关。
3. 输入序列是一个平稳随机过程的样本序列。

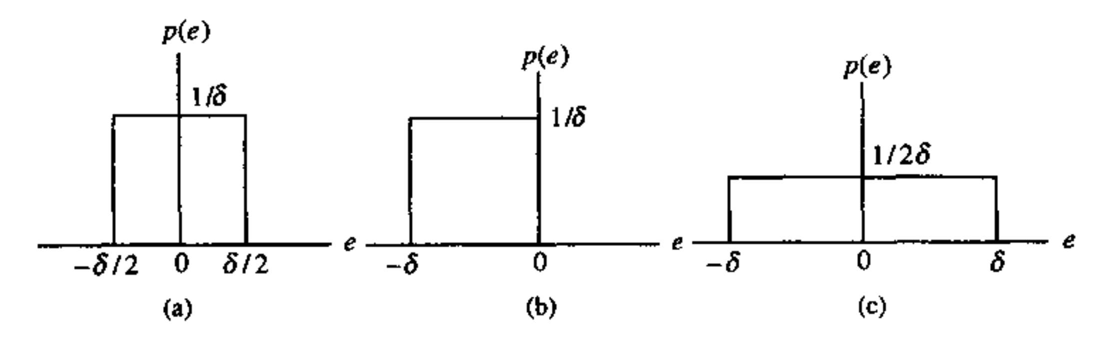


图 9.2 量化误差概率密度函数:(a)舍入,(b)补码截尾,(c)反码截尾

对于长度为$(b+1)$位(其中一位为符号位)的定点小数,$\delta=2^{-b}$。

在反码或原码截尾的情形下,量化误差与输入信号相关,因为每一个误差样本 $e[n]$的符号与相应的输入样本 $x[n]$的符号正好相反。因此采用舍入或补码截尾的量化方式。用舍入量化时,误差样本的均值和方差为

$$m_e=\frac{(\delta/2)-(\delta/2)}{2}=0 \tag{9.1}$$

$$\sigma_e^2=\frac{[(\delta/2)-(-\delta/2)]^2}{12}=\frac{\delta^2}{12} \tag{9.2}$$

用补码截尾量化时,相应的参数如下:

$$m_e=\frac{0-\delta}{2}=-\frac{\delta}{2} \tag{9.3}$$

$$\sigma_e^2=\frac{(0-\delta)^2}{12}=\frac{\delta^2}{12} \tag{9.4}$$

加性量化噪声 $e[n]$对模数转换器的输入信号 $x[n]$的影响,可由量化信噪比($SNR_{A/D}$)给出,定义为

$$SNR_{A/D}=10\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right)\ \text{dB} \tag{9.5}$$

其中 σ_x^2 是输入信号的方差,同时也是输入信号的功率,而 σ_e^2 是噪声的方差,同时也是量化噪声的功率。舍入量化误差在$(-\delta/2,\delta/2)$内服从均匀分布,而补码截尾量化误差在$(-\delta,0)$内服从均匀分布,如图 9.2(a)和图 9.2(b)分别所示。在一个双极$(b+1)$位模数转换器中,$\delta=2^{-(b+1)}R_{FS}$,其中 R_{FS}是转换器的转换幅度范围。因此,

$$SNR_{A/D}=10\log_{10}\left(\frac{48\sigma_x^2}{2^{-2b}(R_{FS})^2}\right)=6.02b+16.81-20\log_{10}\left(\frac{R_{FS}}{\sigma_x}\right)\text{dB} \tag{9.6}$$

R9.5 数字滤波器输出中的噪声是由输入端模数转换器产生的,为了说明这一点,下面分析如图 9.3 所示的模型,其中线性时不变数字滤波器 $H(z)$被认为是无限接近没有量化的

理想情况。数字滤波器的实际输出则由 $y[n]+v[n]$ 给出，式中 $y[n]$ 是由未量化的输入 $x[n]$ 产生的输出，而 $v[n]$ 是由误差序列 $e[n]$ 产生的输出。输出噪声 $v[n]$ 可表示为

$$v[n]=\sum_{m=-\infty}^{\infty}e[m]\,h[n-m] \tag{9.7}$$

其中 $h[n]$ 是数字滤波器的冲激响应。输出噪声 $v[n]$ 的均值 m_v 和方差 σ_v^2 分别可表示为

$$m_v=m_e\,H(e^{j0}), \tag{9.8}$$

$$\sigma_v^2=\frac{\sigma_e^2}{2\pi}\int_{-\pi}^{\pi}|H(e^{j\omega})|^2\,d\omega \tag{9.9}$$

输出噪声功率谱为

$$P_{vv}(\omega)=\sigma_e^2\,|H(e^{j\omega})|^2 \tag{9.10}$$

归一化输出噪声方差为

$$\sigma_{v,n}^2=\frac{\sigma_v^2}{\sigma_e^2}=\frac{1}{2\pi}\int_{-\pi}^{\pi}|H(e^{j\omega})|^2\,d\omega \tag{9.11}$$

$$=\frac{1}{2\pi j}\oint_C H(z)\,H(z^{-1})\,z^{-1}dz \tag{9.12}$$

其中 C 是 $H(z)H(z^{-1})$ 的收敛域中的逆时针方向的曲线。式(9.11)可表示为

$$\sigma_{v,n}^2=\sum_{n=-\infty}^{\infty}|h[n]|^2 \tag{9.13}$$

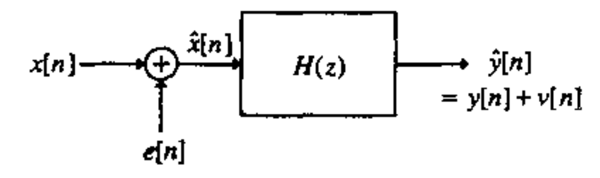


图 9.3　用来分析由一个线性时不变离散时间系统对一个量化的输入处理的效果的分析模型

R9.6　对于具有单极点的因果稳定实有理传输函数 $H(z)$，计算其归一化输出噪声方差时，有一个基于 $H(z)$ 的部分分式形式表示的简单代数方法[Mit74b]：

$$H(z)=\sum_{k=1}^{L}H_k(z) \tag{9.14}$$

其中 $H_k(z)$ 是常数 A 或 $\frac{B_k}{z-z_k}$。将式(9.14)代入式(9.12)，归一化输出噪声方差的表达式可为

$$\sigma_{v,n}^2=\frac{1}{2\pi j}\left\{\sum_{k=1}^{R}\oint_C H_k(z)\,H_k(z^{-1})\,z^{-1}\,dz+2\sum_{k=1}^{R-1}\sum_{\ell=k+1}^{R}\oint_C H_k(z)\,H_\ell(z^{-1})\,z^{-1}\,dz\right\} \tag{9.15}$$

式(9.15)中的曲线积分记为

$$I_i = \frac{1}{2\pi j}\oint_C H_k(z)\,H_\ell(z^{-1})\,z^{-1}\,dz \tag{9.16}$$

不同的情况下,I_i 的表达式会不同,如表 9.1 所示。

表 9.1 典型曲线积分的表示

$H_k(z)$	$H_\ell(z^{-1})$	
	A	$\frac{B_\ell}{z^{-1}-a_\ell}$
A	$I_1 = A^2$	0
$\frac{B_k}{z-a_k}$	0	$I_2 = \frac{B_k B_\ell}{1-a_k a_\ell}$

R9.7 对于因果和稳定的数字滤波器,其冲激响应会很快衰减到零值,因此式(9.13)可近似认为是一个有限项的和:

$$\sigma_{v,n}^2 = \sum_{n=0}^{L} |h[n]|^2 \tag{9.17}$$

确定 L 值的一种实用方法是基于上述对 $L=1,2,\cdots$ 的部分和的递归计算,当 $|h[L]|^2$ 小于一个给定值 k 时,停止计算,k 值通常选为 10^{-7}。

R9.8 在定点数字滤波器中,只有乘法运算的结果会被量化。乘积舍入误差分析的统计模型如图 9.4 所示。在这里,理想乘法器的输出 $v[n]$ 被量化成 $\hat{v}[n]$,其中 $\hat{v}[n]=v[n]+e_a[n]$。

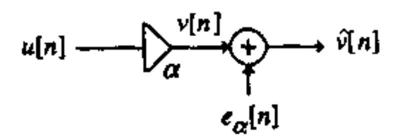


图 9.4 乘积舍入误差分析的统计模型

为便于分析,我们做出如下假定:

1. 误差序列 $\{e_a[n]\}$ 是平稳白噪声随机过程的一个样本序列,其中每一个样本 $e_a[n]$ 在量化误差范围内均匀分布。
2. 量化误差序列 $e[n]$,与序列 $\{v[n]\}$、数字滤波器的输入序列 $\{x[n]\}$ 以及所有其他量化噪声不相关。

   $\{e_a[n]\}$ 与 $\{v[n]\}$ 不相关的假设仅对舍入和补码截尾量化成立。式(9.1)和式(9.2)分别给出了舍入量化的误差样本的均值和方差,而式(9.3)和式(9.4)分别给出了补码截尾量化的误差样本的均值和方差。

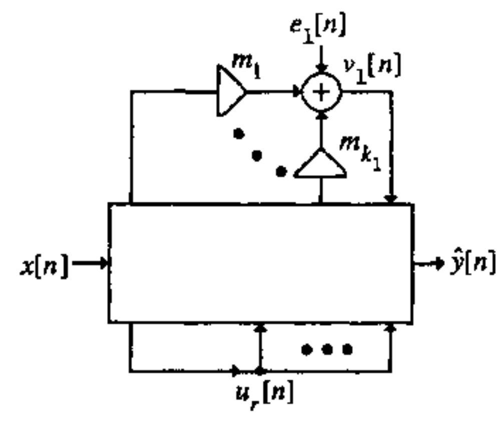


图 9.5 用于乘积舍入误差分析的一个数字滤波器结构的统计模型

R9.9 用于乘积舍入误差分析的一个数字滤波器的统计模型如图 9.5 所示,从图中可以看到

其输入端的 k_ℓ 个乘法器的量化输出。从图中也可以看到内部的第 r 个分支节点,该点的信号变量为 $u_r[n]$,为了防止溢出,该变量需要被截短。误差源在统计意义上假定是相互独立的,因此,数字滤波器的输出的全部舍入噪声是由每一个噪声源产生的噪声的总和。

从第 ℓ 个加法器的输入到数字滤波器输出的冲激响应 $g_\ell[n]$ 的 z 变换 $G_\ell(z)$,称为噪声传输函数,而从数字滤波器输入到第 r 个分支节点的冲激响应 $f_r[n]$ 的 z 变换 $F_r(z)$,称为尺度传输函数。

由噪声源 $e_\ell[n]$ 引起的输出噪声为

$$\sigma_o^2\left[k_\ell\left(\frac{1}{2\pi j}\oint_C G_\ell(z)\,G_\ell(z^{-1})\,z^{-1}\,dz\right)\right]=\sigma_o^2\left[k_\ell\left(\frac{1}{2\pi}\int_{-\pi}^{\pi}|G_\ell(e^{j\omega})|^2\,d\omega\right)\right] \quad (9.18)$$

其中 σ_o^2 表示每个乘法器单独输出的噪声的方差。乘积舍入量化引起的所有输出噪声的功率为

$$\sigma_\gamma^2=\sigma_o^2\sum_{\ell=1}^{L}k_\ell\left(\frac{1}{2\pi j}\oint_C G_\ell(z)\,G_\ell(z^{-1})\,z^{-1}\,dz\right) \quad (9.19)$$

其中 L 是滤波器中加法器的个数。量化运算在乘-加运算之后执行,$k_\ell=1$。

R9.10 在一个定点数字滤波器中,溢出可能在某些内部节点发生,这会导致滤波器输出数据产生较大的振动,从而得到不合要求的结果。在数字滤波器中插入尺度化乘法器来尺度化内部信号级,可在较大程度上减小溢出的可能性。

参考图 9.5,尺度化的目的是确保

$$|u_r[n]|\leqslant 1,\quad \text{对所有 } r \text{ 和所有 } n \text{ 的值成立} \quad (9.20)$$

假定所有定点数都用二进制小数表示,而滤波器的输入序列以单位 1 为界,即

$$|x[n]|\leqslant 1,\text{对所有 } n \text{ 值成立} \quad (9.21)$$

[Jac70]给出了一个通用的尺度化规则:

$$|u_r[n]|\leqslant \|F_r\|_q\cdot\|X\|_p \quad (9.22)$$

对于所有 $p,q\geqslant 1$ 满足 $\frac{1}{p}+\frac{1}{p}=1$。在式(9.22)中,$\|X\|_p$ 表示傅里叶变换 $X(e^{j\omega})$ 的 $\mathcal{L}_p$ 范数($p\geqslant 1$),定义为

$$\|X\|_p\triangleq\left(\frac{1}{2\pi}\int_{-\pi}^{\pi}|X(e^{j\omega})|^p d\omega\right)^{\frac{1}{p}} \quad (9.23)$$

注意,对于 $\mathcal{L}_\infty$ 界,$p=\infty$ 且 $q=1$,而对于 $\mathcal{L}_2$ 界,$p=q=2$。另一个有用的尺度化规则 $\mathcal{L}_1$ 界,对 $p=1$ 和 $q=\infty$ 得到。尺度化之后,尺度化传输函数为 $\|\bar{F}_r\|_q$,选择尺度化常数使得

$$\|\bar{F}_r\|_q\leqslant 1,\qquad r=1,2,\cdots,R \quad (9.24)$$

R9.11 对于已经提出的很多数字滤波器实现技术,它们的系数量化的灵敏度不高。低灵敏度的一个关键原因是给定的传输函数 $H(z)$ 是一个有界实(BR)函数;即 $H(z)$ 是由幅度响应 $|H(e^{j\omega})|$ 描述的因果稳定实系数函数,幅度响应 $|H(e^{j\omega})|$ 的上界为 1,即

$$|H(e^{j\omega})| \leqslant 1 \tag{9.25}$$

也假定在一组频率 ω_k 中,$H(z)$的幅度正好等于 1,即

$$|H(e^{j\omega_k})| = 1 \tag{9.26}$$

由于幅度响应以 1 为上界,频率 ω_k 在滤波器的通带中。若以量化的乘法器系数(假定系数值变换很小)实现的滤波器的传输函数是一个有界实函数,并且满足式(9.25)的条件,而且实现一个有界实传输函数的数字滤波器结构在通带中的这组频率 ω_k 处是零灵敏度,而在通带中的其他频率处灵敏度较小,则这样一种结构就称为结构上无源的。

R9.12 在 R6.19 中讨论过一个有界实传输函数,该函数是并联全通的,它在通带内灵敏度较小,因此是结构上无源的。

有限冲激响应传输函数 $H(z)$是 N 次 1 型有界实函数,它可以通过两个有限冲激响应滤波器 $G_a(z)$和 $G_b(z)$的级联形式表示,其延时互补传输函数 $G(z)$以一个结构上无源的形式实现,其中 $G_b(z)$的全部零点都在单位圆上,且为二重零点:

$$G(z) = G_a(z)G_b(z) = G_a(z)\prod_{k=1}^{L}(1 - 2\cos\omega_k z^{-1} + z^{-2})^2 \tag{9.27}$$

以 $2L$ 个二阶部分的级联实现 $G_b(z)$,其乘法器系数是 $2\cos\omega_k$。传输函数 $H(z)$可用如下形式实现:

$$H(z) = z^{-N/2} - G_a(z)G_b(z) \tag{9.28}$$

如图 9.6 所示。

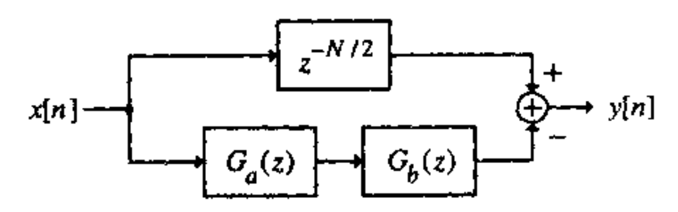


图 9.6 1 型有限冲激响应滤波器 $H(z)$的低通带灵敏度实现

R9.13 实际的数字滤波器是通过数字运算量化实现的非线性系统。因此,在无限精度下稳定的无限冲激响应滤波器,可能在有限精度算法下对特定输入信号(如零或常数输入)不稳定。这类不稳定常常会引起输出数据的极限环振动。极限环有两种类型:(1)粒状和(2)溢出。粒状极限环通常具有低振幅,而溢出极限环振动的振幅较大。

9.3 用到的 MATLAB 命令

这次练习中将使用的 MATLAB 命令如下:

通用命令

```
disp
```

运算符和特殊字符

```
:    .    +    -    *    /    ;    %    <
```

语言构造和调试

```
break    end    for    if    input
```

基本矩阵和矩阵控制

```
ones    pi    zeros
```

基本函数

```
abs    cos
```

多项式和内插函数

```
conv
```

二维图形

```
axis    plot    stem    title    xlabel    ylabel
```

通用图形函数

```
clf    subplot
```

字符串函数

```
num2str
```

信号处理工具箱

```
filter    impz
```

关于这些命令的详细说明，请参阅 *MATLAB Reference Guide*[Mat94]和 *Signal Processing Toolbox User's Guide*[Mat96]，或在命令窗口中键入 help *命令名称*。附录 B 给出了此处使用的 MATLAB 函数的简要解释。

9.4　二进制数的产生和量化

由于 MATLAB 使用十进制数字和算法来研究数字滤波器上的量化效果，而在数字滤波器中采用的是二进制数字和算法，因此用十进制表示二进制数和信号量化更为方便。

例 9.1　与量化二进制数等效的十进制数的产生

此例的目的是将一个原码形式的十进制数转换成二进制数，对于这个二进制数，用给定的位数进行截尾得到小数部分，并将结果表示为十进制。为此，可以用下面给出的 MATLAB 函数 a2dT。

```
function beq = a2dT(d,n)
% BEQ = A2DT(D,N)产生一个
```

```
% 十进制数向量 D 的二进制表示的
% 十进制等效 BEQ,其中 N 位用于
% 通过截尾得到的幅度部分
%
m = 1; d1 = abs(d);
while fix(d1) > 0
    d1 = abs(d)/(10^m);
    m = m + 1;
end
beq = 0;
for k = 1:n
    beq = fix(d1 * 2)/(2^k) + beq;
    d1 = (d1 * 2) - fix(d1 * 2);
end
beq = sign(d). * beq * 10^(m - 1);
```

习题

Q9.1 解释函数 a2dT 完成的运算。命令 fix 的目的是什么？使用函数 a2dT 编写一个 MATLAB 程序，以将一个任意十进制数字转换成为其量化等效形式并显示该等效形式，输入数据是待转换的十进制数 d 和小数部分位数 N，而输出是量化的二进制数的十进制等效形式 beq。

Q9.2 使用上面的程序产生如下十进制数的原码形式的二进制等效：(1)5.3749，(2) - 21.782 39，(3)0.798 89。小数部分有六位。

Q9.3 若小数部分位数为八位，重做习题 Q9.2。

Q9.4 写出在习题 Q9.2 和习题 Q9.3 中产生的二进制数字的反码和补码。

下面给出的 MATLAB 函数 a2dR 将一个十进制数转换成二进制数，对于这个二进制数，用给定的位数舍入得到小数部分，并将结果表示成十进制形式。

```
function beq = a2dTR(d,n)
% BEQ = A2DR(D,N)产生一个
% 十进制数向量 D 的二进制表示的
% 十进制等效 BEQ,其中 N 位用于
% 通过舍入得到的幅度部分
%
m = 1; d1 = abs(d);
while fix(d1) > 0
    d1 = abs(d)/(10^m);
    m = m + 1;
end
beq = 0;d1 = d1 + 2^( - n - 1);
for k = 1:n
    beq = fix(d1 * 2)/(2^k) + beq;
    d1 = (d1 * 2) - fix(d1 * 2);
end
beq = sign(d). * beq * 10^(m - 1);
```

习题

Q9.5 函数 a2dR 和 a2dT 之间有什么不同？舍入是怎样实现的？

9.5 系数量化效应

如前面所示(见R9.4),用量化系数实现的数字滤波器的传输函数通常与所期望的传输函数不同,会引起极点和零点的移动,从而影响到滤波器的频率响应特性偏离设计的要求。在本节中,我们将研究系数量化效应。

例9.2 对频率响应的影响和极零点位置

我们首先研究由无限冲激响应传输函数的直接形式实现的滤波器的系数量化效应。为此,可用到下面给出的程序P9.1。程序使用M函数pzplot,它与M函数zplane基本相同,只是在这里极零点图中极点用符号+表示,而零点用符号*表示[①]。

```
% 程序 P9.1
% 对于无限冲激响应传输函数,用
% 直接型结构实现时的系数量化效应
clf;
[b,a] = ellip(6,0.05,60,0.4);
[g,w] = gain(b,a);
bq = a2dT(b,6);aq = a2dT(a,6);
[gq,w] = gain(bq,aq);
plot(w/pi,g,'b',w/pi,gq,'r--');
axis([0 1 -80 1]);grid
xlabel('\omega/\pi');ylabel('增益,dB');
title('原始的-实线;量化的-虚线');
pause
zplane(b,a);
hold on;
pzplot(bq,aq);
title('原极零点位置: x,o; 新极零点位置: +,*')
```

习题

Q9.6 哪一条语句确定了无限冲激响应传输函数?传输函数的阶数及其类型是什么?哪一条语句确定了传输函数量化系数的十进制形式?对二进制数安排了多少位来表示小数部分?

Q9.7 运行上述程序并产生两个图形。评论你的结果。

Q9.8 修改程序P9.1来研究一个八阶椭圆高通传输函数的系数量化效应,该传输函数的通带波纹为0.1 dB,最小阻带衰减为70 dB,归一化截止频率为0.55弧度/秒。对于二进制数,用五位来表示小数部分。运行修改的程序。评论你的结果。

比较用量化系数实现的一个无限冲激响应传输函数的直接型和级联型的性能是有意义的。程序P9.2反映了级联型实现的乘法器的系数量化效应。

```
% 程序 P9.2
% 对于一个无限冲激响应传输函数,用
```

① 对函数zplane的修改获得了Mathwork公司的许可。

习题

Q9.12 由程序 P9.3 产生的传输函数的阶数及其类型是什么？其要求的幅度响应是什么？

Q9.13 运行上述程序并产生幅度响应图。评论你的结果。

Q9.14 修改程序 P9.3 来研究一个 25 阶等波纹高通传输函数的系数量化效应，该传输函数的阻带范围是从 0 到 0.6，通带范围是从 0.65 到 1。对于二进制数，用四位表示小数部分。运行修改的程序并评论你的结果。

9.6　模数转换噪声分析

例 9.3　模数量化信噪比的求取

在本例中，我们将使用 MATLAB 来研究模数转换器的字长 b 与量化信噪比 $SNR_{A/D}$ 之间的关系。SNR 的计算公式见式(9.6)。

习题

Q9.15 编写一个 MATLAB 程序，用转换幅度范围为 $R_{FS} = K\sigma_x$ 的一个 $(b+1)$ 位模数转换器(其中一位为符号位)来确定零均值高斯分布的模拟样本 $x[n]$ 的量化信噪比。用这个程序分别对 b 值 7,9,11,13,15 和 K 值 4,6 和 8 进行计算。

例 9.4　输出噪声方差的计算

本例研究的是将量化噪声输入到一个因果稳定线性时不变数字滤波器得到输出的传输过程。为此，可使用下面给出的函数 noisepwr1 和 R9.6 中列出的方法。

```
function nvar = noisepwr1(num,den)
% 基于部分分式展开的
% 数字滤波器对于量化输入
% 得到的输出的噪声方差计算
%
% num 和 den 是无限冲激响应传输函数的
% 分子和分母多项式系数
%
[r,p,k] = residue(num,den);
R = size(r,1);
R2 = size(K,1);
if R2 > 1
    disp('不能继续…');
    return;
    end
if R2 = = 1
    nvar = k \ verb + ^ + 2;
    else
    nvar = 0;
    end
% 计算舍入噪声方差
for k = 1:R,
    for m = 1:R,
```

```
        integral = r(k) * conj(r(m))/(1 - p(k) * conj(p(m)));
        nvar = nvar + integral;
        end
end
disp('输出噪声方差 = ');disp(real(nvar));
```

习题

Q9.16 使用函数 noisepwr1 编写一个 MALTAB 程序,以计算具有如下指标的一个四阶椭圆低通滤波器的归一化输出舍入噪声方差:通带波纹为 0.5dB,最小阻带衰减为 50 dB,通带边界在 0.45 处。运行该程序并确定该滤波器的归一化输出舍入噪声方差。

Q9.17 对阻带边界为 0.3 和 0.75 以及最小阻带衰减为 60 dB 的一个六阶 2 型切比雪夫带通滤波器,重做习题 Q9.16。

用式(9.13)来计算归一化输出噪声方差的近似值一般是很方便的(见 R9.7)。下面给出的函数 noisepwr2 就是基于这种方法。

```
function nvar = noisepwr2(num,den)
% 通过冲激响应样本的平方和
% 近似计算基于部分分式展开的
% 一个数字滤波器对量化输入
% 得到的输出的噪声方差
%
% num 和 den 是无限冲激响应传输函数的
% 分子和分母多项式系数
%
x = 1;
order = max(length(num),length(den)) - 1;
si = [zeros(1,order)];
nvar = 0; k = 1;
while k > 0.0000001
    [y,sf] = filter(num,den,x,si);
    si = sf; k = abs(y) * abs(y);
    nvar = nvar + k;
    x = 0;
end
disp('输出噪声方差 = ');disp(nvar)
```

习题

Q9.18 使用函数 noisepwr2 编写一个 MATLAB 程序,以计算一个线性时不变因果稳定数字滤波器的归一化输出舍入噪声方差。用该程序确定习题 Q9.16 中的滤波器的归一化输出噪声方差,并与在习题 Q9.16 中得到的结果进行比较。

9.7 运算舍入误差分析

舍入量化的输出噪声方差可以很容易地由 MATLAB 算出。输出舍入噪声通常只在数字滤波器结构已被尺度化后计算,而在尺度化过程中,系统可能会引入额外的乘法器。实际上,大

多数尺度化乘法器可纳入系统而不明显增加乘法器(噪声源)的总数。对于尺度化后的结构,式(9.19)的输出舍入噪声的表达式为

$$\sigma_{\gamma}^{2}=\sigma_{o}^{2}\sum_{\ell=1}^{L}\bar{k}_{\ell}\left(\frac{1}{2\pi j}\oint_{C}\bar{G}_{\ell}(z)\bar{G}_{\ell}(z^{-1})z^{-1}dz\right) \tag{9.29}$$

其中 $\bar{k}_\ell$ 是反馈到第 ℓ 个加法器的乘法器总数,这里 $\bar{k}_\ell \geqslant k_\ell$,而 $\bar{G}_\ell(z)$ 表示的是从第 ℓ 个加法器的输入开始到滤波器输出之间的噪声传输函数。

$\mathcal{L}_2$ 范数的动态范围尺度化过程可以通过 MATLAB 的数字滤波器结构仿真来实现。若我们将从数字滤波器的输入到第 r 个分支节点的冲激响应以 $\{f_r[n]\}$ 表示,并且假定分支节点按照以 r 的递增的优先关系排序[Cro75],则可首先计算 $\{f_1[n]\}$ 的 $\mathcal{L}_2$ 范数,然后通过乘法器 $k_1 = 1/\|F_1\|_2$ 对输入尺度化。接着,计算 $\{f_2[n]\}$ 的 $\mathcal{L}_2$ 范数并通过除以一个常数 $k_2 = 1/\|F_2\|_2$ 尺度化反馈到第二个加法器的乘法器。这个过程持续到输出节点被尺度化为 1 的 $\mathcal{L}_2$ 范数为止。

例 9.5　级联型无限冲激响应数字滤波器结构

本例中用到了 $\mathcal{L}_2$ 范数,我们将学习级联型无限冲激响应数字滤波器的尺度化,并计算由于尺度化带来的乘积舍入引起的输出噪声方差。

习题

Q9.19　被设计的滤波器是一个具有下列指标的低通椭圆滤波器:通带边界为 0.25,阻带边界为 0.5,通带波纹为 0.5 dB,最小阻带衰减为 50 dB。编写一个 MATLAB 程序来确定每个二阶部分的分子和分母多项式系数。用到的 MATLAB 函数有 `ellipord`,`ellip` 和 `zp2sos`。

图 9.7 所示的是上述传输函数的尺度化级联实现,其中每个部分都使用直接 II 型结构。

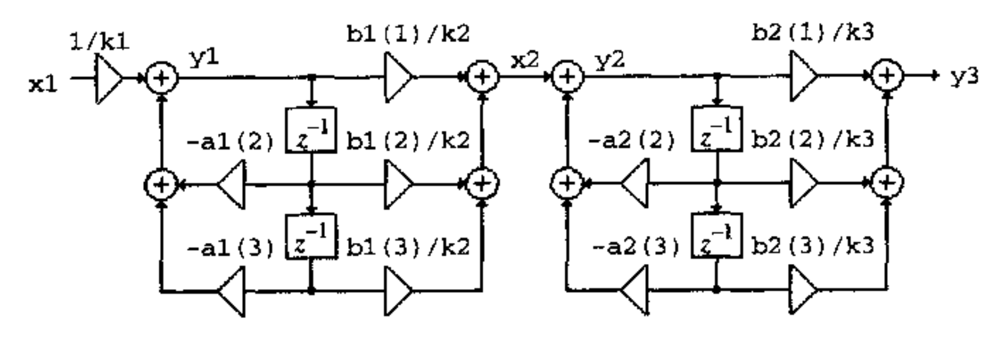


图 9.7　以直接 II 型由习题 Q9.19 产生的传输函数的级联实现

习题

Q9.20　常数 b1(1),b1(2),b1(3),b2(1),b2(2),b2(3),a1(2),a1(3),a2(2)和 a2(3)的值是多少?

该结构的仿真由下面的 MATLAB 程序 P9.4 给出[①]。

① 分母系数中的第一个系数应是 1。

```
% 程序 P9.4
% 图 9.6 所示级联型结构的
% 尺度化和舍入噪声计算
%
b1 = input('第一部分的分子系数 = ');
a1 = input('第一部分的分母系数 = ');
b2 = input('第二部分的分子系数 = ');
a2 = input('第二部分的分母系数 = ');
format long
k1 = 1;
k2 = 1;
k3 = 1;
x1 = 1/k1;
si1 = [0 0]; si2 = [0 0];
var = 0; k = 1;
while k > 0.0000001
    y1 = - a1(2) * si1(1) - a1(3) * si1(2) + x1;
    x2 = (b1(1) * y1 + b1(2) * si1(1) + b1(3) * si1(2))/k2;
    y2 = - a2(2) * si2(1) - a2(3) * si2(2) + x2;
    y3 = (b2(1) * y1 + b2(2) * si1(1) + b2(3) * si1(2))/k3;
    si1(2) = si1(1); si1(1) = y1;
    si2(2) = si2(1); si2(1) = y2;
% 近似的 L2 范数平方计算
    k = abs(y1) * abs(y1);
    var = var + k;
    x1 = 0;
end
disp('L2 范数平方 = ');disp(var);
```

习题

Q9.21 运行程序 P9.4 并确定 y1[n]的$\mathcal{L}_2$范数,这里 x1[n]是一个单位样本序列。现在在程序 P9.4 中设 k1 等于 y1[n]的$\mathcal{L}_2$范数并再次运行它。y1[n]的$\mathcal{L}_2$范数的新值是什么?接下来,修改程序 P9.4 来计算 y2[n]的$\mathcal{L}_2$范数并运行修改后的程序。然后在程序 P9.4 中设置 k2 等于 y2[n]的$\mathcal{L}_2$范数并再次运行它。y2[n]的$\mathcal{L}_2$范数的新值是多少?最后,修改程序 P9.4 来计算 y3[n]的$\mathcal{L}_2$范数并运行修改后的程序。然后在程序 P9.4 中设置 k3 等于 y3[n]的$\mathcal{L}_2$范数并再次运行它。y3[n]的$\mathcal{L}_2$范数的新值是多少?

程序 P9.4 做简单修改后,也可用来确定图 9.7 所示结构的全部输出乘积舍入噪声方差。

习题

Q9.22 若所有乘积被假定在相加前量化,图 9.7 所示结构的噪声模型中,有多少个噪声源进入输出为 y1[n]的加法器?有多少个噪声源进入输出为 y2[n]的加法器?有多少个噪声源进入输出为 y3[n]的加法器?

Q9.23 为了确定单个噪声源进入输出为 y1[n]的加法器而产生的输出乘积舍入噪声方差,在修改了的程序 P9.4 中(保存尺度化常数的所有当前值),用语句"x1 = 1;"代替语句"x1 = 1/k1;"并运行它。为了确定单个噪声源进入输出为 y1[n]的加法器而产生的

输出乘积舍入噪声方差,在修改了的程序 P9.4 中(保存尺度化常数的所有当前值),用语句“x2 = 1;”、语句“x1 = 0;”和语句“x2 = 0;”代替语句“x1 = 1;”并运行它。在所有乘积于相加前被量化以及在相加后被量化两种情况下,输出乘积舍入噪声方差的值各是多少?

Q9.24 将图 9.7 中的第 1 部分和第 2 部分互换,并适当地改变 a1,b1,a2 和 b2 的值,重做习题 Q9.22 和习题 Q9.23。哪一个结构有最低的输出乘积舍入噪声方差?

9.8 低灵敏度数字滤波器

例 9.6 低通带灵敏度无限冲激响应滤波器

在本例中,我们将用 MATLAB 来研究图 6.13 所示并联全通结构的低通带灵敏度,以实现一个有界实无限冲激响应传输函数。

习题

Q9.25 按以下指标使用函数 ellipord 和 ellip 来设计一个奇数阶低通椭圆传输函数:通带边界在 0.4,阻带边界在 0.55,通带波纹为 0.5 dB,最小阻带衰减为 50 dB。用极点交错性质实现并联全通结构(见 R6.19)。在这个结构中需要多少个乘法器? 使用 MATLAB,用非量化和量化的系数(对二进制等效的小数部分用六位表示)实现上述低通滤波器的并联全通结构,得到它们的传输函数的增益响应,并在同一幅图上画出这两个响应。并联的全通结构有低通带灵敏度吗?

例 9.7 低通带灵敏度有限冲激响应滤波器

在本例中,我们将使用 MATLAB 来研究图 9.6 所示延时互补结构滤波器的低通带灵敏度,实现一个 1 型有界实 FIR 传输函数。

习题

Q9.26 使用函数 remez 设计一个长度为 15 的线性相位有限冲激响应滤波器,其归一化通带边界为 0.55,归一化阻带边界为 0.65,通带和阻带波纹有相等的权重。用函数 freqz 计算有限冲激响应滤波器在 0 与 1 之间 1024 个等间隔频率点处的幅度响应,并将滤波器的冲激响应样本除以用命令 max 得到的幅度响应的最大绝对值。尺度化后的传输函数 $H(z)$应是一个有界实函数。通过计算尺度化后的滤波器 $H(z)$的频率响应,验证这个性质。使用函数 a2dT 来量化尺度化后的滤波器 $H(z)$的系数,其中安排五位表示二进制等效的小数部分。在同一幅图上绘制尺度化后的滤波器 $H(z)$和尺度化后的量化滤波器的幅度响应,并评论你的结果。

Q9.27 确定在习题 Q9.26 中尺度化后的有限冲激响应滤波器的延时互补滤波器 $G(z)$的系数。使用命令 roots 确定延时互补滤波器 $G(z)$的零点,并得到由单位圆上的零点组成的因式 $G_b(z)$。使用命令 deconv 通过 $G_b(z)$对 $G(z)$解卷积来确定 $G(z)$剩下的因式 $G_a(z)$。使用函数 a2dT 来量化 $G_a(z)$的系数和 $G_b(z)$的每个因式,其中安排五位表示二进制等效的小数部分。$G(z)$由具有量化系数的 $G_a(z)$和 $G_b(z)$的级联实现,求解 $G(z)$的延时互补滤波器并画出其幅度响应。$H(z)$的这个实现具有低通带灵敏度吗?

9.9 极限环

例 9.8 粒状极限环产生

在本例中,我们将研究由下式给出的一阶无限冲激响应滤波器的粒状极限环的产生:

$$\hat{y}[n] = \mathcal{Q}(\alpha\hat{y}[n-1]) + x[n] \tag{9.30}$$

其中输出 $\hat{y}[n]$是由 $\hat{y}[n-1]$舍入得到的,$x[n]$为输入。为此,可用下面给出的程序 P9.5。

```
% 程序 P9.5
% 一阶无限冲激响应滤波器中的粒状极限环
%
alpha = input('输入 α 的值 = ');
yi = 0; x = 0.04;
for n = 1:21
    y(n) = a2dR(alpha * yi,5) + x;
    yi = y(n); x = 0;
end
k = 0:20;
stem(k,y)
ylabel('振幅'); xlabel('时间序号 n')
```

习题

Q9.28 函数 a2dR 的目的是什么? 对量化的二进制小数部分安排了多少位? 对下列 α 值运行上述程序:-0.55 和 0.55。评论你的结果。

Q9.29 改变量化二进制小数部分的位数,重做习题 Q9.28。

Q9.30 对于滤波器系数 α 的不同值,重做习题 Q9.28。评论你的结果。

例 9.9 溢出极限环产生

在本例中,我们将研究由下式给出的二阶无限冲激响应滤波器的溢出极限环的产生:

$$\hat{y}[n] = \mathcal{Q}(-\alpha_1\hat{y}[n-1] - \alpha_2\hat{y}[n-2]) + x[n] \tag{9.31}$$

其中输出 $\hat{y}[n]$是对 $-\alpha_1\hat{y}[n-1]$与 $-\alpha_2\hat{y}[n-2]$之和舍入得到的,而 $x[n]$是输入。为此,可用下面给出的程序 P9.6。

```
% 程序 P9.6
% 二阶无限冲激响应滤波器中的溢出极限环
%
alpha = input('α1 和 α2 值 = ');
yi1 = 0.75; yi2 = -0.75;
for n = 1:41
    y(n) = - alpha(1) * yi1 - alpha(2) * yi2;
    y(n) = a2dR(y(n),3);
    yi2 = yi1; yi1 = y(n);
end
k = 0:40;
stem(k,y)
ylabel('振幅'); xlabel('时间序号 n')
```

习题

Q9.31 对量化二进制小数部分安排了多少位？对下列滤波器系数值运行上述程序：$\alpha_1 = -0.875$和 $\alpha_2 = 0.875$。评论你的结果。这个极限环与习题 Q9.28 产生的极限环有什么不同？

Q9.32 改变对量化二进制小数部分的位数，重做习题 Q9.31。

Q9.33 用滤波器系数 α_1 与 α_2 的不同值，重做习题 Q9.31。评论你的结果。

Q9.34 在程序 P9.6 中，通过用函数 `a2dT` 代替 `a2dR`，重做习题 Q9.31。在程序 P9.6 中，$-\alpha_1\hat{y}[n-1]$与$-\alpha_2\hat{y}[n-2]$的和是通过截尾来量化的。评论你的结果。

9.10 基础知识阅读

[1] A. Antoniou. *Digital Filters: Analysis, Design, and Applications.* McGraw-Hill, New York NY, second edition, 1993. Ch. 11.

[2] E. Cunningham. *Digital Filtering: An Introduction.* Houghton-Mifflin, Boston MA, 1992. Ch. 8.

[3] D.J. DeFatta, J.G. Lucas, and W.S. Hodgkiss. *Digital Signal Processing: A System Design Approach.* Wiley, New York NY, 1988. Ch. 9.

[4] L.B. Jackson. *Digital Filters and Signal Processing.* Kluwer, Boston MA, third edition, 1996. Ch. 11.

[5] R. Kuc. *Introduction to Digital Signal Processing.* McGraw-Hill, New York NY, 1988. Ch. 10.

[6] S.K. Mitra. *Digital Signal Processing: A Computer-Based Approach.* McGraw-Hill, New York NY, 1998. Sec. 8.4, Ch. 9.

[7] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing.* Prentice-Hall, Englewood Cliffs NJ, 1989, Secs. 6.7 – 6.10.

[8] S.J. Orfanidis. *Introduction to Signal Processing.* Prentice-Hall, Englewood Cliffs NJ, 1996, Ch. 2, Sec. 7.6.

[9] B. Porat. *A Course in Digital Signal Procesing.* Wiley, New York NY, 1996. Sec. 11.4 – 11.8.

[10] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications.* Prentice-Hall, Englewood Cliffs NJ, third edition, 1996. Sec. 1.4, Secs. 7.5 – 7.7.

[11] R.A. Roberts and C.T. Mullis. *Digital Signal Processing.* Addison-Wesley, Reading MA, 1987. Ch. 9.

第 10 章 多抽样率数字信号处理

10.1 导言

由于在系统输入端和输出端以及所有内部节点的抽样率相同,迄今为止我们所讨论的数字信号处理结构都属于单抽样率系统。但是有时在系统的不同部分(包括输入端和输出端)采用不同的抽样率是有必要的,而且通常非常方便。在本实验练习中,我们首先将学习使用 MALTAB 的上抽样器和下抽样器特性,它们是一个多抽样率系统的基本成分,然后将它们应用到更复杂系统(如内插器、抽取器和滤波器组)的设计中。

10.2 基础知识回顾

R10.1 图 10.1 所示的上抽样器以整数因子 $L>1$ 来增加序列 $x[n]$ 的抽样率。它在序列 $x[n]$ 的连续两个抽样点之间插入 $L-1$ 个等距的零值抽样点来生成输出序列 $x_u[n]$:

$$x_u[n]=\begin{cases} x[n/L], & n=0,\pm L,\pm 2L,\cdots \\ 0, & \text{其他} \end{cases} \tag{10.1}$$

上抽样器是一个线性时变的离散时间系统。在 z 域中,其输入输出关系为

$$X_u(z)=X(z^L) \tag{10.2}$$

其中 $X(z)$ 和 $X_u(z)$ 分别表示 $x[n]$ 和 $x_u[n]$ 的 z 变换。对于 $z=e^{j\omega}$,上面的等式变为 $X_u(e^{j\omega})=X(e^{j\omega L})$。

$x[n] \longrightarrow \boxed{\uparrow L} \longrightarrow x_u[n]$

图 10.1 上抽样器

R10.2 图 10.2 所示的下抽样器以整数因子 $M>1$ 来减少序列 $x[n]$ 的抽样率。它根据关系式

$$y[n]=x[nM] \tag{10.3}$$

去掉抽样点间的 $M-1$ 个点来产生输出序列 $y[n]$。

$x[n] \longrightarrow \boxed{\downarrow M} \longrightarrow y[n]$

图 10.2 下抽样器

下抽样器也是一个线性时变的离散时间系统。在 z 域中,其输入输出关系为

$$Y(z)=\frac{1}{M}\sum_{k=0}^{M-1}X(z^{1/M}W_M^{-k}) \tag{10.4}$$

其中 $X(z)$ 和 $X_u(z)$ 分别表示 $x[n]$ 和 $y[n]$ 的 z 变换。对于 $z=e^{j\omega}$,上式变为

$$Y(e^{j\omega})=\frac{1}{M}\sum_{k=0}^{M-1}X(e^{j(\frac{\omega-2\pi k}{M})})\tag{10.5}$$

由式(10.5)可知,当且仅当信号 $x[n]$被限在 $\pm\pi/M$ 之间时,才不会存在由于因子 M 下抽样所引起的混叠。

R10.3 当且仅当 M 和 L 互质,即 M 和 L 没有一个整数公因子 $k>1$ 时,级联的 M 因子下抽样器和 L 因子上抽样器(见图 10.3)可在输入输出关系不变的情况下互换。

图 10.3　上抽样器和下抽样器的级联

R10.4 图 10.4 给出了两种其他的级联等效情况。

(a)

(b)

图 10.4　级联等效

R10.5 由于上抽样器引起了 $x[n]$谱的周期重复,上抽样后信号 $x_u[n]$的谱中产生了多余的镜像,但这些镜像可通过在上抽样操作之后加入低通滤波器 $H(z)$来消除(如图 10.5(a)所示)。图 10.5(a)所示的系统称为内插器。低通内插滤波器 $H(z)$的幅频特性为

$$|H(e^{j\omega})|=\begin{cases}L, & |\omega|\leqslant\omega_c/L\\ 0, & \pi/L\leqslant|\omega|\leqslant\pi\end{cases}\tag{10.6}$$

其中 ω_c 表示待内插信号中需要保留的最高频率。

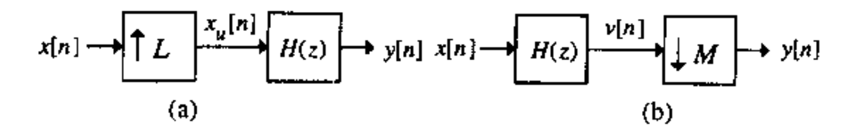


图 10.5　(a)内插器,(b)抽取器

R10.6 类似地,在下抽样之前信号 $v[n]$必须通过低通滤波器 $H(z)$将带宽限制到 $|\omega|<\pi/M$ 来消除由下抽样操作引起的混叠。图 10.5(b)所示的系统称为抽取器。低通抽取滤波器 $H(z)$的幅频特性为

$$|H(e^{j\omega})|=\begin{cases}1, & |\omega|\leqslant\omega_c/M\\ 0, & \pi/M\leqslant|\omega|\leqslant\pi\end{cases}\tag{10.7}$$

R10.7 得到分数变化的抽样率构造如图 10.6 所示,其中低通滤波器 $H(z)$含有归一化阻带截止频率

$$\omega_s = \min\left(\frac{\pi}{L}, \frac{\pi}{M}\right) \tag{10.8}$$

图 10.6 分数率抽样率改变方案

R10.8 通常将变抽样率系统设计成几个部分的级联,可使得计算效率明显增加。图 10.7(a)是 L 因子内插器的两级设计,其中 $L = L_1 L_2$。同样地,图 10.7(b)是 M 因子抽取器的两级设计,其中 $M = M_1 M_2$。

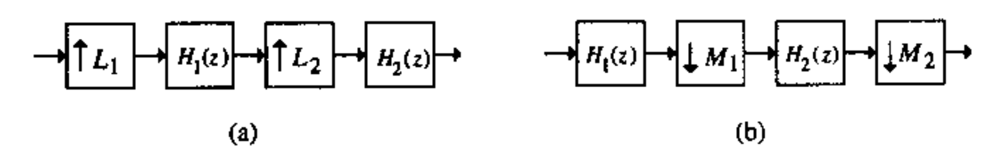


图 10.7 两级实现:(a)内插器,(b)抽取器

R10.9 任意序列$\{x[n]\}$的 z 变换 $X(z)$

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n} \tag{10.9}$$

的 M 频带多相分解为

$$X(z) = \sum_{k=0}^{M-1} z^{-k} X_k(z^M) \tag{10.10}$$

其中,

$$X_k(z) = \sum_{n=-\infty}^{\infty} x_k[n] z^{-n} = \sum_{n=-\infty}^{\infty} x[Mn+k] z^{-n}, \quad k = 0, 1, \cdots, M-1 \tag{10.11}$$

子序列$\{x_k[n]\}$称为原序列$\{x[n]\}$的多相成分,而$\{x_k[n]\}$的 z 变换函数 $X_k(z)$称为 $X(z)$的多相成分。子序列$\{x_k[n]\}$和原序列$\{x[n]\}$之间的关系为

$$x_k[n] = x[Mn+k], \quad k = 0, 1, \cdots, M-1 \tag{10.12}$$

R10.10 N 阶有限冲激响应传输函数 $H(z)$的 I 型多相分解为

$$H(z) = \sum_{k=0}^{M-1} z^{-k} E_k(z^M) \tag{10.13}$$

而其 II 型多相分解为

$$H(z) = \sum_{\ell=0}^{M-1} z^{-(M-1-\ell)} R_\ell(z^M) \tag{10.14}$$

其中,

$$R_\ell(z) = E_{M-1-\ell}(z), \quad \ell = 0, 1, \cdots, M-1 \tag{10.15}$$

基于式(10.13)和式(10.14)分解的 $H(z)$的直接实现如图 10.8 所示。

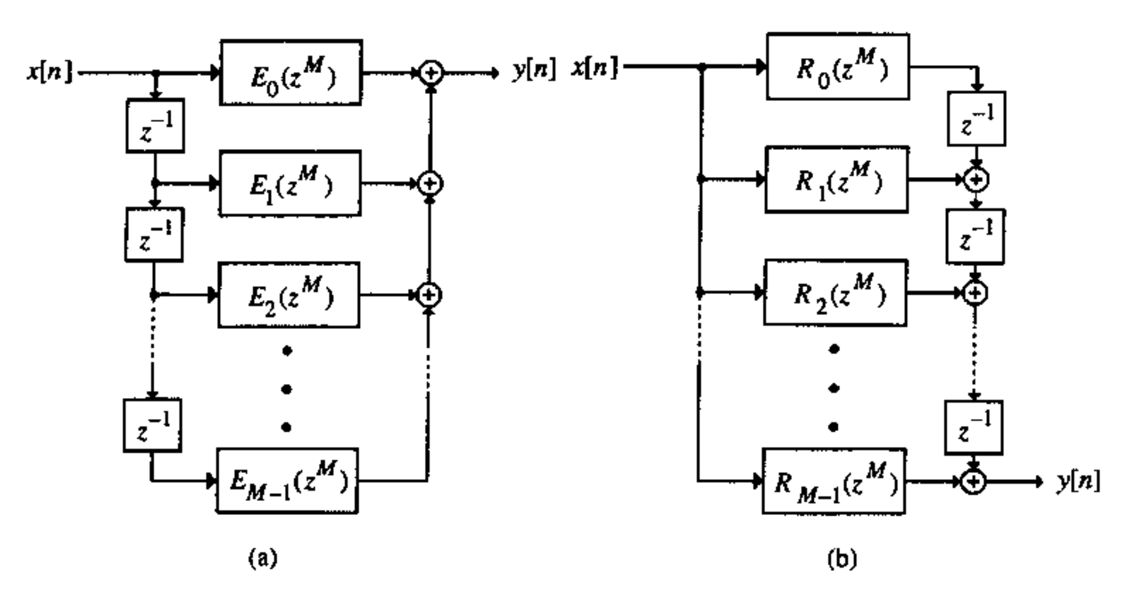


图 10.8 基于一个多相分解的有限冲激响应滤波器实现:(a) I 型多相分解,(b) II 型多相分解

R10.11 数字滤波器组是如图 10.9 所示的带有共同输入或相加输出的一组数字带通滤波器。图 10.9(a)所示的结构称为 M 频带分析滤波器组,其中子滤波器 $H_k(z)$是我们所知的分析滤波器。同样地,图 10.9(b)所示的结构称为 L 频带合成滤波器组,其子滤波器 $F_k(z)$是我们所知的合成滤波器。

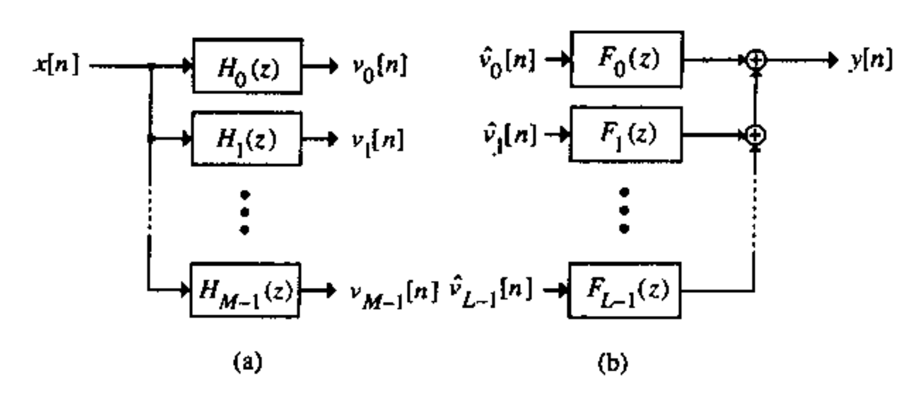


图 10.9 (a)分析滤波器组,(b)合成滤波器组

R10.12 设 $H_0(z)$表示具有冲激响应为 $h_0[n]$的因果低通数字滤波器:

$$H_0(z) = \sum_{n=0}^{\infty} h_0[n]\, z^{-n} \tag{10.16}$$

其中通带边界为 ω_p,阻带边界为 ω_s,$\omega_p < \pi/M < \omega_s$,$M$ 为任意整数。设传输函数 $H_k(z)$为

$$\begin{aligned} H_k(z) &= \sum_{n=0}^{\infty} h_k[n]\, z^{-n} = \sum_{n=0}^{\infty} h_0[n]\,(z\,W_M^k)^{-n} \\ &= H_0(z\,W_M^k), \quad k = 0, 1, \cdots, M-1 \end{aligned} \tag{10.17}$$

其中 $W_M = e^{-j2\pi/M}$。这些传输函数的频率响应为

$$H_k(e^{j\omega}) = H_0(e^{j(\omega - \frac{2\pi k}{M})}), \quad k = 0, 1, \cdots, M-1 \tag{10.18}$$

因此,$H_k(z)$的频率响应可通过将 $H_0(z)$的响应以步长 $2\pi k/M$ 右移得到。式(10.17)定义的 M 个滤波器 $H_k(z)$可在图 10.9(a)所示的分析滤波器组中作为分析滤波器使用,或在图 10.9(b)所示的合成滤波器组中作为合成滤波器使用。得到的滤波器组称为均匀滤波器组。

R10.13 若低通原型传输函数 $H_0(z)$以其 M 频带多相形式表示:

$$H(z) = \sum_{\ell=0}^{M-1} z^{-\ell} E_\ell(z^M) \tag{10.19}$$

其中 $E_\ell(z)$是 $H_0(z)$的第 ℓ 个多相成分,具体为

$$E_\ell(z) = \sum_{n=0}^{\infty} e_\ell[n] z^{-n} = \sum_{n=0}^{\infty} h_0[\ell + nM] z^{-n}, \quad 0 \leqslant \ell \leqslant M-1 \tag{10.20}$$

M 个滤波器 $H_k(z)$, $k = 0, 1, \cdots, M-1$ 可用矩阵形式表达为

$$\begin{bmatrix} H_0(z) \\ H_1(z) \\ H_2(z) \\ \vdots \\ H_{M-1}(z) \end{bmatrix} = M\,\mathbf{D}_M^{-1} \begin{bmatrix} E_0(z^M) \\ z^{-1}E_1(z^M) \\ z^{-2}E_2(z^M) \\ \vdots \\ z^{-(M-1)}E_{M-1}(z^M) \end{bmatrix} \tag{10.21}$$

其中 $\mathbf{D}_M$ 表示 $M \times M$ 离散傅里叶变换矩阵:

$$\mathbf{D}_M = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_M^1 & W_M^2 & \cdots & W_M^{M-1} \\ 1 & W_M^2 & W_M^4 & \cdots & W_M^{2(M-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_M^{(M-1)} & W_M^{2(M-1)1} & \cdots & W_M^{(M-1)^2} \end{bmatrix} \tag{10.22}$$

因此,基于式(10.21)的 M 频带分析滤波器组的有效实现框图如图 10.10 所示,其中原型低通滤波器 $H_0(z)$已用其多相形式表示。图 10.10 所示的结构更一般地称为均匀傅里叶变换分析滤波器组。均匀傅里叶变换合成滤波器组相应的多相实现可用类似的方法得到,如图 10.11 所示。

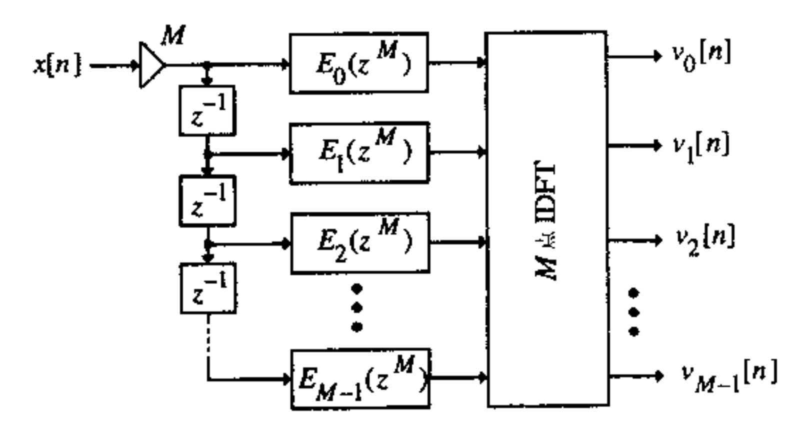


图 10.10　均匀傅里叶变换分析滤波器组

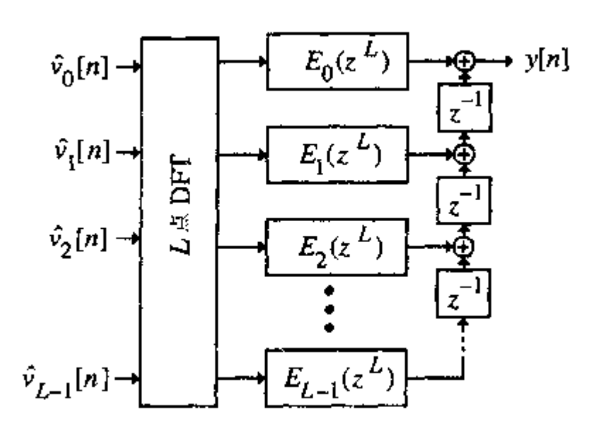


图 10.11　均匀傅里叶变换合成滤波器组

R10.14　通带边界 ω_p 和阻带边界 ω_s 满足关系 $\omega_p < \pi/M < \omega_s$。且冲激响应 $h[n]$ 满足

$$h[Ln] = \begin{cases} \alpha, & n = 0 \\ 0, & 其他 \end{cases} \tag{10.23}$$

的低通滤波器 $H(z)$ 称为 L 频带滤波器或奈奎斯特滤波器。

对于这样一个滤波器，多相分解形如

$$H(z) = \alpha + \sum_{\ell=1}^{L-1} z^{-\ell} E_\ell(z^L) \tag{10.24}$$

和

$$\sum_{k=0}^{L-1} H(zW_L^k) = L\alpha = 1, \quad (假设\ \alpha = \frac{1}{L}) \tag{10.25}$$

R10.15　当 $L = 2$ 时，L 频带滤波器 $H(z)$ 称为半带滤波器，它的传输函数为

$$H(z) = \alpha + z^{-1}E_1(z^2) \tag{10.26}$$

$L = 2$ 时，其冲激响应满足式(10.23)。式(10.25)给出的频率响应的条件在这里简化为

$$H(z) + H(-z) = 1 \quad (假设\ \alpha = \frac{1}{2}) \tag{10.27}$$

若 $H(z)$ 有实系数，则 $H(-e^{j\omega}) = H(e^{j(\pi-\omega)})$，且由式(10.27)可推出

$$H(e^{j\omega}) + H(e^{j(\pi-\omega)}) = 1 \tag{10.28}$$

线性相位有限冲激响应半带滤波器 $H(z)$ 的冲激响应 $h[n]$ 的长度限制为 $4K+3$ 形式（除非 $H(z)$ 是一个常数）。

10.3　用到的 MATLAB 命令

本章练习中将使用的 MATLAB 命令如下：

通用命令

```
length
```

运算符和特殊符号

```
:    .    +    -    *    /    ;
%    .*   =
```

语言构造和调试

```
end    for
```

基本矩阵和矩阵控制

```
pi    sum
```

基本函数

```
abs    exp    log10    sin
```

二维图形

```
axis     clf     grid     plot     stem
subplot  title   xlabel   ylabel
```

信号处理工具箱

```
decimate    fir2        freqz    hamming    interp
remez       resample    sinc
```

关于这些命令的详细说明,请参阅 *MATLAB Reference Guide*[Mat94]和 *Signal Processing Toolbox User's Guide*[Mat96],或在命令窗口中键入 help *命令名称*。附录 B 给出了此处使用的 MATLAB 函数的简要解释。

10.4 基本抽样率改变器件

本节的目的是学习在时域和在频域中使用 MATLAB 上抽样及下抽样操作。

例 10.1 时域中的输入输出关系

程序 P10.1 可用于研究上抽样操作。

```
%程序 10.1
%以一个整数因子进行上抽样
%
clf;
n = 0:50;
x = sin(2 * pi * 0.12 * n);
y = zeros(1,3 * length(x));
y([1: 3: length(y)]) = x;
subplot(2,1,1)
stem(n,x);
title('输入序列');
xlabel('时间序号 n');ylabel('振幅');
subplot(2,1,2)
stem(n,y(1:length(x)));
```

```
title('输出序列');
xlabel('时间序号 n');ylabel('振幅');
```

习题

Q10.1　正弦序列中以弧度为单位的角频率是多少？其长度是多少？上抽样因子 L 是多少？

Q10.2　在程序 P10.1 中，上抽样操作是怎样实现的？

Q10.3　运行程序 P10.1，验证输出和输入序列之间的关系满足式(10.1)。

Q10.4　分别取两个不同的角频率值和上抽样因子 L 值，重做习题 Q10.3。

Q10.5　修改程序 P10.1，研究对一个斜坡序列的上抽样操作。

程序 P10.2 可用于研究下抽样操作。

```
% 程序 10.2
% 以一个整数因子进行下抽样
%
clf;
n = 0: 49;
m = 0: 50 * 3 - 1;
x = sin(2 * pi * 0.042 * m);
y - x([1: 3: length(x)]);
subplot(2,1,1)
stem(n,x(1:50)); axis([0 50 - 1.2 1.2]);
title('输入序列');
xlabel('时间序号 n');
ylabel('振幅');
subplot(2,1,2)
stem(n,y); axis([0 50 - 1.2 1.2]);
title('输出序列');
xlabel('时间序号 n');
ylabel('振幅');
```

习题

Q10.6　正弦序列中以弧度为单位的角频率是多少？其长度是多少？下抽样因子 M 是多少？

Q10.7　在程序 P10.2 中，下抽样操作是怎样实现的？

Q10.8　运行程序 P10.2，验证输出和输入序列之间的关系满足式(10.3)。

Q10.9　分别取两个不同的角频率值和下抽样因子 M 值，重做习题 Q10.8。

例 10.2　频域中的输入输出关系

为了在频域中说明上抽样和下抽样的效果，我们需要构造一个在时域中有限长而同时在频域中有限带宽的输入序列。为此可使用 M 函数 fir2(见 7.5 节)。

程序 P10.3 可以用来研究上抽样器的频域特性。

```
% 程序 P10.3
% 频域中上抽样的效果
% 使用 fir2 构造一个带限输入序列
clf;
```

```
freq=[0 0.45 0.5 1];
mag=[0 1 0 0];
x=fir2(99,freq,mag);
%求取并画出输入谱
[Xz,w]=freqz(x,1,512,'whole');
subplot(2,1,1);
plot(w/pi,abs(Xz)); axis([0 1 0 1]); grid
xlabel('\omega/\pi'); ylabel('幅度');
title('输入谱');
subplot(2,1,2);
%产生上抽样的序列
L=input('输入上抽样因子 - ');
y=zeros(1,L*length(x));
y([1: L: length(y)])=x;
%求取并画出输出谱
[Yz,w]=freqz(y,1,512,'whole');
plot(w/pi,abs(Yz)); axis([0 1 0 1]); grid
xlabel('\omega/\pi'); ylabel('幅度');
title('输出谱');
```

习题

Q10.10 输入序列的长度是多少？其幅度谱是什么？

Q10.11 取下面的上抽样因子值，运行程序 P10.3：$L=2,3$ 和 5。讨论由程序产生的图形。其结果是否与式(10.2)一致？

程序 P10.4 可用于研究下抽样的频域特性。

```
%程序 P10.4
%频域中下抽样的效果
%使用 fir2 构造一个带限输入序列
clf;
freq=[0 0.42 0.48 1]; mag=[0 1 0 0];
x=fir2(101,freq,mag);
%求取并画出输入谱
[Xz,w]=freqz(x,1,512);
subplot(2,1,1);
plot(w/pi,abs(Xz)); grid
xlabel('\omega/\pi'); ylabel('幅度');
title('输入谱');
%产生下抽样的序列
M=input('输入下抽样因子 - ');
y=x([1: M: length(x)]);
%求取并画出输出谱
[Yz,w]=freqz(y,1,512);
subplot(2,1,2);
plot(w/pi,abs(Yz)); grid
xlabel('\omega/\pi'); ylabel('幅度');
title('输出谱');
```

习题

Q10.12 输入序列的长度是多少？其幅度谱是什么？

```
clf;
L = input('上抽样因子 = ');
% 产生输入序列
n = 0:49;
x = sin(2 * pi * 0.043 * n) + sin(2 * pi * 0.031 * n);
% 产生内插后的输出序列
y = interp(x,L);
% 画出输入和输出序列
subplot(2,1,1);
stem(n,x(1:50));
title('输入序列');
xlabel('时间序号 n'); ylabel('振幅');
subplot(2,1,2);
m = 0:(50 * L) - 1;
stem(m,y(1:50 * L));
title('输出序列');
xlabel('时间序号 n'); ylabel('振幅');
```

习题

Q10.18 形成输入序列的两个正弦序列的频率是多少？输入序列的长度是多少？

Q10.19 内插滤波器的类型和阶数是多少？

Q10.20 取 $L=2$，运行程序 P10.6 并讨论结果。

Q10.21 将输入信号中两个正弦序列的频率改为 0.045 和 0.029，输入序列的长度改为 40，对 $L=3$，运行修改后的程序 P10.6 并讨论你的结果。

例 10.5 分数率抽样率改变

程序 P10.7 使用 M 函数 resample 设计和实现具有分数内插因子 L/M 的内插器。函数 resample 设计并利用具有阻带边界的低通内插滤波器来满足式(10.8)。

```
% 程序 P10.7
% 以两个整数的比
% 改变抽样率
%
clf;
L = input('上抽样因子 = ');
M = input('下抽样因子 = ');
n = 0:29;
x = sin(2 * pi * 0.043 * n) + sin(2 * pi * 0.031 * n);
y = resample(x,L,M);
subplot(2,1,1);
stem(n,x(1:30));axis([0 29 -2.2 2.2]);
title('输入序列');
xlabel('时间序号 n'); ylabel('振幅');
subplot(2,1,2);
m = 0:(30 * L/M) - 1;
stem(m,y(1:30 * L/M));axis([0 (30 * L/M) - 1 -2.2 2.2]);
title('输出序列');
xlabel('时间序号 n'); ylabel('振幅');
```

习题

Q10.22 形成输入序列的两个正弦序列的频率是多少？输入序列的长度是多少？

Q10.23 带限滤波器的类型和阶数是多少？

Q10.24 取 $L=5$ 和 $M=3$，运行程序 P10.7 并讨论结果。

Q10.25 将输入信号中两个正弦序列的频率改为 0.045 和 0.029，输入序列长度改为 40，对 $L=3$ 和 $M=5$，运行修改后的程序 P10.7 并讨论你的结果。

10.6 滤波器组的设计

例 10.6 均匀分析/合成滤波器组

利用式(10.17)可很容易地实现 M 频带均匀分析或合成滤波器组的设计，式中 $\{h_0[n]\}$ 是具有通带边界 ω_p 和阻带边界 ω_s，$\omega_p<\pi/M<\omega_s$ 的原型低通滤波器。该设计可用程序 P10.8 来实现。

```
% 程序 P10.8
% 均匀离散傅里叶变换滤波器组的设计
clf;
% 设计原型低通滤波器
b = remez(20,[0 0.2 0.25 1],[1 1 0 0],[10 1]);
w = 0:2 * pi/255:2 * pi; n = 0:20;
for k = 1:4;
    c = exp(2 * pi * (k - 1) * n * i/4)
    FB = b. * c;
    HB(k,:) = freqz(FB,1,w);
end
% 画出每个滤波器的幅度响应
subplot(2,2,1)
plot(w/pi,abs(HB(1,:)));
xlabel('\omega/\pi');ylabel('振幅');
title('滤波器 No.1'); axis([0 2 0 1.1]);
subplot(2,2,2)
plot(w/pi,abs(HB(2,:)));
xlabel('\omega/\pi');ylabel('振幅');
title('滤波器 No.2'); axis([0 2 0 1.1]);
subplot(2,2,3)
plot(w/pi,abs(HB(3,:)));
xlabel('\omega/\pi');ylabel('振幅');
title('滤波器 No.3'); axis([0 2 0 1.1]);
subplot(2,2,4)
plot(w/pi,abs(HB(4,:)));
xlabel('\omega/\pi');ylabel('振幅');
title('滤波器 No.4'); axis([0 2 0 1.1]);
```

习题

Q10.26 程序 P10.8 中原型低通滤波器的类型和阶数是多少？其指标是什么？滤波器组中有多少个频带？

Q10.27 运行程序 P10.8 并讨论你的结果。

Q10.28 修改程序 P10.8 来设计一个三频带均匀滤波器组,画出它们的幅度响应。使用相同的原型低通滤波器

10.7 奈奎斯特滤波器的设计

例 10.7 加窗傅里叶级数法

通过 R7.9 中所描述的加窗傅里叶级数法,可以很容易地设计并得到具有良好频率响应且截止频率在 $\omega_c=\pi/L$ 处的低通线性相位奈奎斯特 L 频带有限冲激响应滤波器。在本方法中,低通滤波器的冲激响应系数选择为

$$h[n]=h_{LP}[n]\cdot w[n] \tag{10.29}$$

其中 $h_{LP}[n]$是截止频率在 π/L 处的理想低通滤波器的冲激响应,而 $w[n]$是一个合适的窗函数。若

$$h_{LP}[n]=0 \quad \text{for} \;\; n=0,\pm L,\pm 2L,\cdots \tag{10.30}$$

则满足式(10.23)

将 $\omega_c=\pi/L$ 代入式(7.14),可得理想 L 频带滤波器的冲激响应 $h_{LP}[n]$为

$$h_{LP}[n]=\frac{\sin(\pi n/L)}{\pi n}, \quad -\infty\leqslant n\leqslant\infty \tag{10.31}$$

由上述情况可以看出,冲激响应系数的确满足式(10.23)给出的条件。因此,通过对式(10.31)应用一个适当的窗函数就可设计一个 L 频带滤波器。

程序 P10.9 可用于实现 L 频带低通滤波器的设计。

```
%程序 P10.9
%用加窗傅里叶级数法
%设计 L 频带有限冲激响应滤波器
%
clf;
K = 11;
n = -K:K;
%产生理想低通滤波器的
%截短冲激响应
b = sinc(n/2)/2;
%产生窗函数
win = hamming(23);
%产生加窗滤波器的系数
fil = b.*win';
c = fil/sum(fil);
%画出加窗滤波器的增益响应
[h,w] = freqz(c,1,256);
g = 20*log10(abs(h));
plot(w/pi,g);axis([0 1 -90 10]); grid
xlabel('\omega/\pi');ylabel('增益,dB');
```

习题

Q10.29 由程序 P10.9 设计的 L 频带有限冲激响应滤波器的长度和 L 的值是多少？程序中使用的是什么类型的窗函数？

Q10.30 运行程序 P10.9 并讨论你的结果。显示 L 频带滤波器的系数并验证其满足式(10.23)。

Q10.31 对不同的 L 值及不同的窗函数，重做习题 Q10.30。

10.8 基础知识阅读

[1] R.E. Crochiere and L.R. Rabiner. *Multirate Digital Signal Processing*. Prentice-Hall, Englewood Cliffs NJ, 1996. Chs. 2, 4, 7.

[2] D.J. DeFatta, J.G. Lucas, and W.S. Hodgkiss. *Digital Signal Processing: A System Design Approach*. Wiley, New York NY, 1988. Ch. 7.

[3] S.K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, New York NY, 1998. Sec. 8.4 and Ch. 10.

[4] S.J. Orfanidis. *Introduction to Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1996, Ch. 12.

[5] B. Porat, *A Course in Digital Signal Procesing*. John Wiley, New York NY, 1996. Ch. 12.

[6] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs NJ, third Edition, 1996. Ch. 10.

[7] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs NJ, 1993. Chs. 4, 5.

第 11 章　高 级 主 题

11.1　导言

本章的练习中包含了 12 个用 MATLAB 实现的例子。在这些例子中，我们使用了在前 10 章中已经学过的数字信号处理概念和工具。在开始之前，我们需要认真地进行计划，并可能需要阅读一些本书没有提及到的其他资料。通过这些例子，我们将进一步理解数字信号处理的理论及算法的应用。另外，我们还将了解这方面的一些新进展。

11.2　离散变换

例 11.1　子带离散傅里叶变换

长度为 N 的序列 $x[n]$，$0\leqslant n\leqslant N-1$，其 N 点离散傅里叶变换为 $X[k]$，$k=0,1,\cdots,N-1$，$X[k]$ 可由在单位圆上的等间隔点的 z 变换 $X(z)=\sum_{n=0}^{N-1}x[n]z^{-n}$ 给出：

$$X[k]=X(z)\Big|_{z=W_N^{-k}},=\sum_{n=0}^{N-1}x[n]W_N^{kn},\quad 0\leqslant k\leqslant N-1 \tag{11.1}$$

其中 $W_N=e^{-j2\pi/N}$。子带离散傅里叶变换(SB-DFT)算法，可用于近似计算长度为 2 的整数次幂的序列的频率范围的一个或多个部分的离散傅里叶变换。为了说明此方法的基本思想，首先按照

$$\begin{aligned}g_L[n]&=\frac{1}{2}\left(x[2n]+x[2n+1]\right)\\ g_H[n]&=\frac{1}{2}\left(x[2n]-x[2n+1]\right),\quad 0\leqslant n\leqslant\frac{N}{2}-1\end{aligned} \tag{11.2}$$

将 $x[n]$分解成为两个长度为$\frac{N}{2}$的子序列 $g_L[n]$和 $g_H[n]$。注意原序列 $x[n]$可由

$$\begin{aligned}x[2n]&=g_L[n]+g_H[n]\\ x[2n+1]&=g_L[n]-g_H[n],\quad 0\leqslant n\leqslant\frac{N}{2}-1\end{aligned} \tag{10.3}$$

从上述子序列中恢复。

$x[n]$的 z 变换 $X(z)$现在可表示为

$$X(z)=(1+z^{-1})G_L(z^2)+(1-z^{-1})G_H(z^2) \tag{11.4}$$

其中 $G_L(z)$和 $G_H(z)$分别是 $g_L[n]$和 $g_H[n]$的 z 变换。$x[n]$的 N 点离散傅里叶变换$X[k]$可另外表示为

$$X[k]=(1+W_N^k)G_L[\langle k\rangle_{N/2}]+(1-W_N^k)G_H[\langle k\rangle_{N/2}],\quad 0\leqslant k\leqslant N-1 \tag{11.5}$$

其中，$G_L[k]$和 $G_H[k]$分别是子序列 $g_L[n]$和 $g_H[n]$的 $N/2$ 点离散傅里叶变换。式(11.5)给出的

离散傅里叶变换算法称为子带离散傅里叶变换[She95]。注意,两个 $N/2$ 点离散傅里叶变换可由任何快速傅里叶变换算法得到。

式(11.5)可用矩阵形式写为

$$X[k]=[1\quad W_N^k]\mathbf{R}_2^{-1}\begin{bmatrix}G_L[\langle k\rangle_{N/2}]\\G_H[\langle k\rangle_{N/2}]\end{bmatrix}\tag{11.6}$$

其中 $\mathbf{R}_2=\begin{bmatrix}1&1\\1&-1\end{bmatrix}$是 2×2 哈达玛矩阵。重复该过程,得到一个二级算法为

$$X[k]=[1\quad W_N^k\quad W_N^{2k}\quad W_N^{3k}]\mathbf{R}_4^{-1}\begin{bmatrix}G_{LL}[\langle k\rangle_{N/4}]\\G_{LH}[\langle k\rangle_{N/4}\\G_{HL}[\langle k\rangle_{N/4}]\\G_{HH}[\langle k\rangle_{N/4}\end{bmatrix}\tag{11.7}$$

其中,$\mathbf{R}_4$ 是 4×4 哈达玛矩阵,而 $G_{LL}[\langle k\rangle_{N/4}]$等是 $N/4$ 点离散傅里叶变换。该步骤可以继续,直到所有的离散傅里叶变换长度为 2。若 $N=2^\mu$,则 μ 级子带离散傅里叶变换算法所需要的复数乘法总数量,与任何库利-图基类型快速傅里叶变换算法所需要的数量相同,但是加法的次数更多。

通过消除可忽略能量的频带的计算,可以简化离散傅里叶变换的算法。例如,对一个基本上是低频率成分的序列 $x[n]$,假定 $x[n]$是一种实值序列,通过从式(11.5)中消除 $G_H[k]$的贡献,可得到一个近似的离散傅里叶变换:

$$X[k]\approx(1+W_N^k)G_L[\langle k\rangle_{N/2}],\qquad 0\leqslant k\leqslant\frac{N}{4}-1\tag{11.8}$$

$X[k]$的高频部分假定是零值。使用式(11.8)所示近似算法的计算量是式(11.5)所示算法的计算量的一半。依据序列 $x[n]$的谱特性,在一个 M 级分解中,任何一个或多个 N/M 的离散傅里叶变换,可以用子带离散傅里叶变换计算方案近似地求取。

编写 MATLAB 程序来确定基于二级子带离散傅里叶变换计算方案的一个序列的离散傅里叶变换。用 M 函数 `fir2` 产生一个带限谱限制在范围 $0\leqslant\omega\leqslant\pi/4$ 的长度为 64 的实序列。用子带离散傅里叶变换程序确定其近似的离散傅里叶变换。比较子带离散傅里叶变换的低频部分与用函数 `fft` 得到的离散傅里叶变换的精确低频部分。用命令 `flops` 来比较这两个离散傅里叶变换计算方案的复杂度。

接下来,修改上述程序,以确定基于 4 级子带离散傅里叶变换计算方案的一个序列的离散傅里叶变换。用新的子带离散傅里叶变换程序计算长度为 64 的序列的近似离散傅里叶变换。比较由 4 级子带离散傅里叶变换与由 2 级子带离散傅里叶变换产生的低频部分。用命令 `flops` 来比较这两个子带离散傅里叶变换计算方案的复杂度。评价你的结果。

例 11.2　非均匀离散傅里叶变换

长度为 N 的序列$x[n]$的 N 点非均匀离散傅里叶变换(NDFT)$X_{NDFT}[k]$定义为[Bag98]

$$X_{NDFT}[k]=\sum_{n=0}^{N-1}x[n]\,z_k^{-n},\qquad 0\leqslant k\leqslant N-1\tag{11.9}$$

其中 $z_0,z_1,\cdots,z_{N-1}$是在 z 平面上任意位置的 N 个不同点。上面的 N 个方程可用矩阵形式表示为

$$\begin{bmatrix} X_{NDFT}[0] \\ X_{NDFT}[1] \\ \vdots \\ X_{NDFT}[N-1] \end{bmatrix} = \mathbf{D}_N \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \tag{11.10}$$

其中,

$$\mathbf{D}_N = \begin{bmatrix} 1 & z_0^{-1} & z_0^{-2} & \cdots & z_0^{-(N-1)} \\ 1 & z_1^{-1} & z_1^{-2} & \cdots & z_1^{-(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & z_{N-1}^{-1} & z_{N-1}^{-2} & \cdots & z_{N-1}^{-(N-1)} \end{bmatrix} \tag{11.11}$$

是 $N\times N$ 非均匀离散傅里叶变换矩阵。注意,对 $z_k=e^{j2\pi k/N}$,$0\leqslant k\leqslant N-1$,非均匀离散傅里叶变换简化为由式(11.1)给出的离散傅里叶变换。式(11.11)给出的非均匀离散傅里叶变换矩阵是一个范德蒙矩阵,且对于每一个 z_k 它的逆惟一。接下来用

$$\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} = \mathbf{D}_N^{-1} \begin{bmatrix} X_{NDFT}[0] \\ X_{NDFT}[1] \\ \vdots \\ X_{NDFT}[N-1] \end{bmatrix} \tag{11.12}$$

来求解式(11.10),可得到序列 $x[n]$的值。

由于范德蒙矩阵通常不够准确,且可能得到大量的数值错误,所以上面所示的非均匀离散傅里叶逆变换算法在实际中不推荐使用。非均匀离散傅里叶逆变换可以通过使用某些多项式内插法(如拉格朗日内插法)对给定的 N 点非均匀离散傅里叶变换 $X[k]$求取 z 变换 $X(z)$来计算:

$$X(z)=\sum_{n=0}^{N-1} x[n]\,z^{-n} \tag{11.13}$$

在该方法中,$X(z)$为

$$X(z)=\sum_{n=0}^{N-1} \frac{I_k(z)}{I_k(z_k)} X_{NDFT}[k] \tag{11.14}$$

其中,

$$I_k(z)=\prod_{\substack{i=0 \\ i\neq k}}^{N-1} \left(1-z_i z^{-1}\right) \tag{11.15}$$

写出一个 M 函数,在 z 平面单位圆上给定的一组不同的 N 个点 z_k,计算一个序列的 N 点非均匀离散傅里叶变换。函数的输入是频率点 z_k和输入序列,而输出是通过非均匀离散傅里叶变换得到的频域值。用拉格朗日内插法写出另一个 M 函数来求非均匀离散傅里叶逆变换。用这些函数计算不同长度的序列的非均匀离散傅里叶变换及其非均匀离散傅里叶逆变换。

非均匀离散傅里叶变换的一个非常好的应用是,用修正频率抽样法设计一维和二维有限冲激响应滤波器,以及设计具有给定零陷的波束形成器[Bag98]。

例 11.3 弯曲离散傅里叶变换

如式(11.1)所示,长度为 N 的序列 $x[n]$的 N 点离散傅里叶变换 $X[k]$,由在单位圆中 N 个

等间隔点 $x[n]$ 的 z 变换 $X(z)$ 的频域值给出。弯曲离散傅里叶变换(WDFT)是用一个弯曲的频率尺度确定 $X(z)$ 的 N 个频域值。$x[n]$ 的 N 点弯曲离散傅里叶变换 $\check{X}[k]$，可通过对 $X(z)$ 应用全通一阶谱变换得到的改进 z 变换 $X(\hat{z})$ 的单位圆中 N 个等间隔的频域值给出[Mit98b]：

$$X(\hat{z}) = X(z)\Big|_{z^{-1}=\frac{-\alpha+\hat{z}^{-1}}{1-\alpha\hat{z}^{-1}}} \tag{11.16}$$

式中 $|\alpha|<1$。因此，$x[n]$ 的 N 点弯曲离散傅里叶变换 $\check{X}[k]$ 为

$$\check{X}[k] = X(\hat{z})\,|_{\hat{z}=e^{j2\pi k/N}},\quad 0\leqslant k\leqslant N-1 \tag{11.17}$$

此时，改进的 z 变换 $X(\hat{z})$ 为

$$X(\hat{z}) = \sum_{n=0}^{N-1} x[n]\left(\frac{-\alpha+\hat{z}^{-1}}{1-\alpha\hat{z}^{-1}}\right)^{-n} = \frac{P(\hat{z})}{D(\hat{z})} \tag{11.18}$$

其中，

$$P(\hat{z}) = \sum_{n=0}^{N-1} p[n]\,\hat{z}^{-n} = \sum_{n=0}^{N-1} x[n]\,(1-\alpha\hat{z}^{-1})^{N-1-n}(-\alpha+\hat{z}^{-1})^{n} \tag{11.19}$$

而

$$D(\hat{z}) = \sum_{n=0}^{N-1} d[n]\,\hat{z}^{-n} = (1-\alpha\hat{z}^{-1})^{N-1} \tag{11.20}$$

注意，多项式 $D(\hat{z})$ 仅仅依赖弯曲参数 α，而多项式 $P(\hat{z})$ 依赖输入 $x[n]$ 和 α。由上可知 N 点弯曲离散傅里叶变换 $\check{X}[k]$ 由 $P[k]/D[k]$ 给出，其中 $P[k]$ 和 $D[k]$ 分别是式(11.19)和式(11.20)定义的序列 $p[n]$ 和 $d[n]$ 的 N 点离散傅里叶变换。

若我们记 $\mathbf{P}=[p[0]\ p[1]\ \cdots\ p[N-1]]^T$ 和 $\mathbf{X}=[x[0]\ x[1]\ \cdots\ x[N-1]]^T$，则可表示为 $\mathbf{P}=\mathbf{Q}\cdot\mathbf{X}$。其中 $\mathbf{Q}=[q_{r,s}]$ 是一个实 $N\times N$ 矩阵，其第一行为 $q_{0,s}=\alpha^s$，$0\leqslant s\leqslant N-1$，而第一列为 $q_{r,0}={}^{N-1}C_r\alpha^r$。剩下的元素 $q_{r,s}$ 可由下面的递归关系得到：

$$q_{r,s} = q_{r-1,s-1} + \alpha\, q_{r,s-1} - \alpha\, q_{r-1,s} \tag{11.21}$$

弯曲离散傅里叶变换有几种有意义的应用。例如，在不改变序列长度 N 的情况下，在角频率轴的给定部分，可以选取弯曲参数增加频率分辨率，如有必要，可在指定频率点确定频域值。具有适当弯曲参量的短长度弯曲离散傅里叶变换可用来提供必要的谱分辨率，这在对包括空间正弦信号在内的一些信号的谱分析中是很有吸引力的。弯曲离散傅里叶变换的另一个应用是设计可调幅度响应有限冲激响应滤波器。

编写一个 MATLAB 函数 `wdft`，以计算一个序列的弯曲离散傅里叶变换。函数的输入数据是输入序列向量和弯曲参数 α，而输出是弯曲离散傅里叶变换抽样的向量。当 $\alpha=-0.45$ 时，使用此函数对信号的不同点数的离散傅里叶变换进行谱分析，信号由频率为 0.35 和 0.65(弧度)的两个正弦信号合成。两个正弦曲线所需要的弯曲离散傅里叶变换的最小点数 N_1 是多少？现在，用函数 `fft` 进行同样信号的谱分析。两个正弦曲线所需要的离散傅里叶变换的最小点数 N_1 是多少？比较基于弯曲离散傅里叶变换和基于离散傅里叶变换方法的计算复杂度。

作为第二个应用，编写一个 MATLAB 程序，设计并绘制具有给定通带边界的低通有限冲激响应滤波器的幅度响应，以及变换后的具有不同通带边界的低通有限冲激滤波器的幅度响应。

利用函数 kaiord(见 7.5 节)来估计有限冲激响应滤波器的长度 N,并用函数 remez 设计原有限冲激响应滤波器,对给定的弯曲参数,计算其幅度响应的弯曲离散傅里叶变换。这些频域值的离散傅里叶逆变换则产生变换后的滤波器的冲激响应。你的程序的输入数据是具有给定通带边界的低通有限冲激响应滤波器的通带边界和阻带边界、其长度及弯曲参数。假定在指定频带权相等 对输入数据的不同值运行你的程序。评论你的结果。

11.3 有限冲激响应滤波器的设计和实现

由于有限冲激响应滤波器很稳定而且无极限环,因此它的应用比无限冲激响应滤波器更为广泛。然而,一个给定幅度响应的有限冲激响应滤波器的长度总是与过渡带宽成反比,因此,在尖锐截止滤波器的情况下,滤波器的长度通常很大而需要大量的乘法和加法运算。在这一节中,我们将研究运算效率较高的某些特定类型的有限冲激响应滤波器的设计。

例 11.4 内插有限冲激响应滤波器

内插有限冲激响应滤波器的设计方法是基于传输函数为 $F(z^L)$ 和 $G(z)$ 的两个有限冲激响应滤波器的级联实现的,得到总的传输函数为

$$H(z) = F(z^L)G(z) \tag{11.22}$$

滤波器 $F(z^L)$ 的频率响应是以 $2\pi/L$ 为周期的。滤波器 $G(z)$ 只保留频率响应需要的通带,不需要的会被衰减。在时域中,$F(z^L)$ 的冲激响应是在模型滤波器 $F(z)$ 的相邻的冲激响应样本之间插入 $L-1$ 个零值得到的。内插滤波器 $G(z)$ 则对 $F(z^L)$ 的冲激响应做内插并补零。

内插有限冲激响应方法可以用来设计窄带低通、高通和带通滤波器[Neu84]。我们在此首先考虑设计一个通带边界在 ω_p、阻带边界在 ω_s、通带波纹为 δ_p、阻带波纹为 δ_s 的窄带低通滤波器 $H(z)$。为此,首先设计一个通带边界在 $L_{\omega p}$、阻带边界在 $L_{\omega s}$、通带波纹为 δ_p、阻带波纹为 δ_s 的宽带低通滤波器 $F(z)$。注意,L 的最大值为$\lfloor \pi/\omega_s \rfloor$。实际上,$L$ 应该选择得比该值稍小,这样可以在 $F(z^L)$ 重复的通带之间提供一些间隔,使得 $G(z)$ 的设计更为简单。为了衰减 $\omega=\pi$ 周期延拓的部分,可以选择有限冲激响应 $G_0(z)=\frac{1}{2}(1+z^{-1})$ 并使中心在 ω_o 正中间的周期延拓的部分得到衰减。在选定 K 值使得增益最大 0 dB 处,可以采用有限冲激响应 $G_2(z)=\frac{1}{K}(1-2\cos\omega_o z^{-1}+z^{-2})$。为了使不需要的通带中的衰减更大,可用到上面两个有限冲激响应的倍数。注意,一旦挑选了 L 和 $G(z)$ 的形式,若要补偿 $G(z)$ 的幅度响应 $|G(e^{j\omega})|$ 在级联的幅度响应上的效果,可以通过 $|G(e^{j\omega})|$ 的逆改变 $H(z)$ 需要的通带带内的 $F(z)$ 的幅度响应来实现。

用函数 remez,设计一个长度为 99、通带边界在 $\omega_p=0.0404\pi$ 处、阻带边界在 $\omega_s=0.0556\pi$ 处且在通带和阻带中权重相等的内插有限冲激响应低通滤波器。对于 $L=2$ 的情况,先设计长度为 49 的模型滤波器 $F(z)$,并预补偿 $G(z)$ 的效果。比较用函数 remez 设计的内插有限冲激响应滤波器及有限冲激响应滤波器的幅度响应和计算复杂度。其次,使用 $L=3$ 和 4 来设计内插有限冲激响应滤波器。并与 $L=2$ 的内插限冲激响应滤波器的性能进行比较。用第 9 章的函数 a2dT,比较用函数 remez 设计的有限冲激响应滤波器和内插有限冲激响应滤波器的系数灵敏度。

通过简单的修改，上面列出的设计方法可用于设计内插有限冲激响应高通滤波器和内插有限冲激响应带通滤波器。例如，若 $H(z)$ 是模型低通有限冲激响应滤波器，通过在 $\omega=0$ 处的内插器 $G_1=(1-z^{-1})$ 来衰减 $H(z^2)$ 在该处的周期延拓，可以设计一个高通滤波器。用这种修改，设计通带边界 $\omega_p=0.9596\pi$ 处、阻带边界在 $\omega_s=0.94454\pi$ 处且通带和阻带权重相等的窄带高通有限冲激响应滤波器。

还可以通过设计的一个窄带延时互补内插有限冲激响应滤波器，来设计一个宽带有限冲激响应滤波器。然而，在这种情况下，宽带滤波器必须是一个 I 型线性相位滤波器。对于内插有限冲激响应滤波器方法的细节，见[Neu84]。

内插有限冲激响应方法已被延用到设计运算效率较高的有限冲激响应带通滤波器[Neu87]。此方法是基于调制模型低通滤波器 $H(z)$ 的冲激响应 $h[n]$ 来产生通带中心分别在 ω_o 和 $-\omega_o$ 的两个复系数带通有限冲激响应滤波器 $H_1(z)$ 和 $H_2(z)$ 来实现的。$H_1(z^L)$ 和 $H_2(z^L)$ 的不需要的通带则分别由复系数有限冲激响应内插器 $G_1(z)$ 和 $G_2(z)$ 衰减。对于实系数 $H(z)$，$H_2(z^L)$ 的系数是 $H_1(z^L)$ 的复共轭。同样地，$G_2(z)$ 的系数也是 $G_1(z)$ 的系数的复共轭。令 $H(z)=H_{re}(z)+jH_{im}(z)$ 及 $G_1(z)=G_{re}(z)+jG_{im}(z)$，其中 $H_{re}(z)$，$H_{im}(z)$，$G_{re}(z)$ 及 $G_{im}(z)$ 有实系数，则所求的有限冲激响应带通滤波器的传输函数为 $\frac{1}{2}\{H_{re}(z)G_{re}(z)+H_{im}(z)G_{im}(z)\}$，得到一个并联结构。

用上面列出的方法，编写一个 MATLAB 程序来设计中心频率在 0.87π 处、通带宽度为 0.02π、过渡带宽为 0.045π 的对称有限冲激响应带通滤波器。通带波纹小于或等于 ±0.025 dB，最小阻带衰减是 51 dB[Neu87]。画出该滤波器的幅度响应，并将其与用函数 `remez` 设计的一个有限冲激响应带通滤波器的图形做比较。比较两个滤波器的硬件复杂性。

例 11.5 频率响应掩蔽法

频率响应掩蔽法可用来设计具有尖锐过渡带的线性相位有限冲激响应滤波器[Lim86]。该方法的基本思想如下。

假设 $F_a(z)$ 是频率响应为

$$F_a(e^{j\omega})=e^{-n\omega/2}R(\omega) \tag{11.23}$$

的 N 阶 1 型低通有限冲激响应滤波器，其中 $R(\omega)$ 是其振幅响应。设 $\omega_{a,p}$ 和 $\omega_{a,s}$ 分别表示 $F_a(z)$ 的通带和阻带边界。$F_a(z)$ 的延时互补滤波器 $F_c(z)$ 为

$$F_c(z)=z^{-N/2}-F_a(z) \tag{11.24}$$

它有一个振幅响应[$1-R(\omega)$]。现在考虑图 11.1(a)所示的结构，其中滤波器 $F_a(z^M)$ 是一个有 M 个通带的多通带滤波器。同样，$F_c(z^M)$ 也是振幅响应与 $F_a(z^M)$ 互补的多通带滤波器。$F_a(z^M)$ 的任何一个或多个通带可用掩蔽滤波器 $G_a(z)$ 掩蔽，而 $F_c(z^M)$ 的一个或多个通带同样也可用掩蔽滤波器 $G_c(z)$ 掩蔽。恰当地选择 $G_a(z)$ 和 $G_c(z)$，通过增加图 11.1(b)所示的 $F_a(z^M)G_a(z)$ 和 $F_c(z^M)G_c(z)$ 的级联的输出，可设计一个具有尖锐过渡带的宽带线性相位有限冲激响应滤波器。例如，若 $G_a(z)$ 的通带和阻带边界在 $(2m\pi+\omega_{a,p})/M$ 和 $[2(m+1)\pi-\omega_{a,s}]/M$ 处，而 $G_c(z)$ 的通带和阻带边界在 $(2m\pi-\omega_{a,p})/M$ 和 $(2m\pi+\omega_{a,s})/M$ 处，则图 11.1(b)所示的结

构实现了通带和阻带边界在

$$\omega_p=\frac{2m\pi+\omega_{a,p}}{M},\quad \omega_s=\frac{2m\pi+\omega_{a,s}}{M} \tag{11.25}$$

处的宽带低通滤波器,其中 m 是一个小于 M 的整数。注意,此处默认 $G_a(z)$ 和 $G_c(z)$ 的群延迟是相等的。若它们不相等,则要增加较小的群延迟使它们的群延迟相等。而且,为了避免半样本延时,NM 必须是偶数。这种方法的详细内容请参阅[Lim86]。

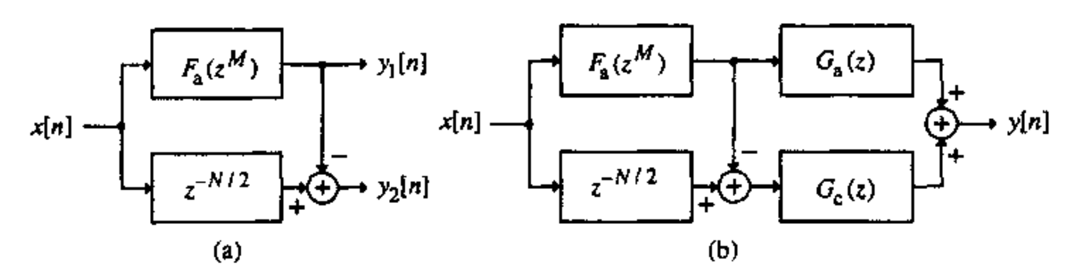


图 11.1 (a)延时互补对,(b)应用频率掩蔽法的有限冲激响应滤波器结构

用 MATLAB,按如下指标基于频率掩蔽法设计一个线性相位有限冲激响应滤波器:通带边界在 0.3π 处,阻带边界在 0.305π 处,通带波纹为 ±0.1 dB,最小阻带衰减为 40 dB。

例 11.6 以 0, +1 和 -1 为系数的有限冲激响应滤波器

为了简化硬件实现,以 0, +1 和 -1 为系数的有限冲激响应滤波器的设计已被提出。在这样的一种设计方法中,整个滤波器的结构如图 11.2 所示[Bat80]。这种方法的基本思想是通过一种类似 δ 调制的方法将常规的有限冲激响应(FIR)滤波器 $H(z)$ 的系数转换为 0, +1 和 -1。为此,对 $H(z)$ 的冲激响应{$[h[n]$}进行 L 倍上抽样,得到相应的冲激响应为{$[h_L[n]$}的传输函数 $H_L(z)=H(z^L)$。接下来的目的是确定样本值为 0, +1 和 -1 的函数{$w[n]$},其游动和在均方误差意义下是序列{$[h_L[n]$}的一个良好逼近,即误差

$$\mathcal{E}_L=\sum_{n=0}^{\infty}\left(h_L[n]-\Delta_L\sum_{\ell=0}^{n}w[\ell]\right)^2 \tag{11.26}$$

在给定的容限内是最小的。在式(11.26)中,Δ_L 是在 δ 调制中相应于步长的尺度化因子。这种方法和最优算法的详细内容,请参阅[Bat80]。

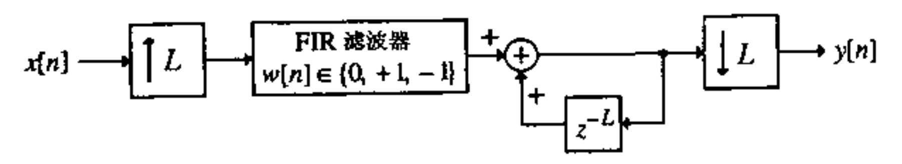


图 11.2 系数为 0, +1 和 -1 的有限冲激响应滤波器实现

基于上面的方法,用 MATLAB 设计一个通带边界在 0.007π 处、阻带边界在 0.023π 处的低通滤波器。采用 6 倍上抽样。

例 11.7 游动有限冲激响应滤波器结构

对于运算效率较高的有限冲激响应滤波器实现,许多作者提出了采用多抽样率技术实现

的不同滤波器结构。图 11.3 所示的就是其中的一种[Vet88]。分析这个结构并证明它是无混叠的，且整个传输函数为

$$T(z)=z^{-1}\left[H_0(z^2)+z^{-1}H_1(z^2)\right] \tag{11.27}$$

对于一个给出的传输函数 $T(z)$，可得到两个传输函数 $H_0(z^2)$和 $H_1(z^2)$如下：

$$\begin{aligned} H_0(z^2) &= \frac{1}{2}[T(z)+T(-z)] \\ H_1(z^2) &= \frac{1}{2}z[T(z)-T(-z)] \end{aligned} \tag{11.28}$$

若 $T(z)$的长度是 $2K$，则两个滤波器 $H_0(z)$和 $H_1(z)$的长度均为 K。

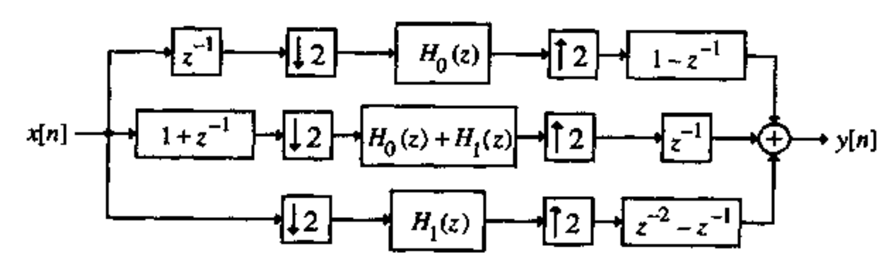


图 11.3 快速游动卷积结构

编写一个 MATLAB 程序，以从一个给定的有限冲激响应传输函数确定传输函数 $H_0(z)$和 $H_1(z)$，然后仿真图 11.3 所示的结构。用 R8.1 中给出的结构验证方法来验证仿真的正确性。使用例 8.1 中描述的函数 strucver。

11.4 滤波器组的应用

在前一章的 R10.11 至 R10.13 中，我们提到过数字分析滤波器及合成滤波器组有许多实际的应用。在此将研究两个这样的应用

例 11.8 多路复用器

多路复用器如图 11.4 所示，其输入端是一个合成滤波器组，输出端是一个分析滤波器组，它们形成了一个有 L 个输入、L 个输出的多抽样率结构。设计多路复用器的目的是，确保对于所有 k 值，第 k 个输入 $x_k[n]$的合理输出为 $y_k[n]$。若 $x_k[n]$包含来自 $x_r[n]$的贡献，则第 k 个和第 r 个信道有串话，其中 $r\neq n$。在完全重构多路复用器中，$y_k[n]=\alpha_k x_k[n-D]$，其中 α_k 是一个常数，而 D 是一个正整数。多路复用器用于时分复用(TDM)到频分复用(FDM)的形式转换。关于多路复用器的理论、设计和应用方面的其他细节，请参阅[Mit98a]。

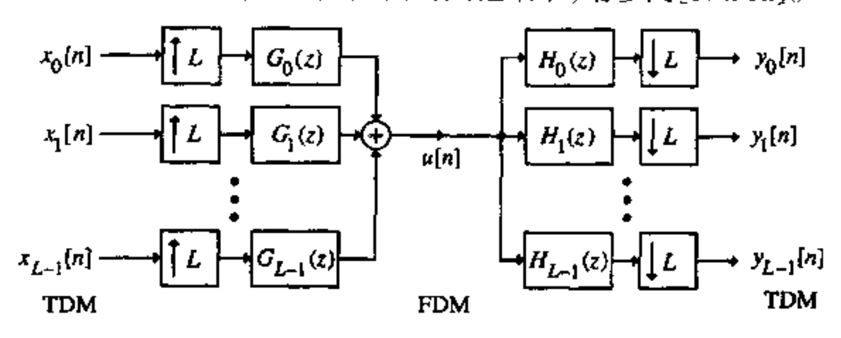


图 11.4 L 通道多路复用器

在本例中,我们将研究具有下列合成及分析滤波器的双通道多路复用器的运算[Mit98a]:

$$G_0(z) = z^{-1} + z^{-2}, \qquad G_1(z) = z^{-1} - z^{-2}$$

$$H_0(z) = 1 + z^{-1}, \qquad H_1(z) = 1 - z^{-1}$$

编写一个 MATLAB 程序来仿真这个结构并处理长度均为 20 的任意两个输入序列。证明在 $k = 0, 1$ 时,$y_k[n] = 2x_k[n-2]$。

例 11.9 正交镜像滤波器组

图 11.5 所示的是 L 通道正交镜像滤波器(QMF)组,它是由输入端的一个 L 通道分析滤波器组和输出端的一个 L 通道合成滤波器组组成的。分析滤波器的下抽样输出信号为 $u_k[n]$,称为子带信号,它的抽样率比正交镜像滤波器组的输入更低,因此可以更有效地处理。可以设计分析滤波器以及合成滤波器,使得正交镜像滤波器组是无混叠的,输入为 $x[n]$,输出为 $y[n]$。正交镜像滤波器组的常见应用是语音、音频、图像和视频信号的子带编码。在本例中,我们将研究图 11.6 所示的一个双通道正交镜像滤波器组的设计和运算。分解该结构并证明分析滤波器和合成滤波器为

$$H_0(z) = \frac{1}{2}\{\mathcal{A}_0(z^2) + z^{-1}\mathcal{A}_1(z^2)\}, \quad H_1(z) = \frac{1}{2}\{\mathcal{A}_0(z^2) - z^{-1}\mathcal{A}_1(z^2)\} \tag{11.29}$$

$$G_0(z) = \frac{1}{2}\{\mathcal{A}_0(z^2) + z^{-1}\mathcal{A}_1(z^2)\}, \quad G_1(z) = \frac{1}{2}\{z^{-1}\mathcal{A}_1(z^2) - \mathcal{A}_0(z^2)\} \tag{11.30}$$

接下来证明在 z 域中,这个正交镜像滤波器组的输入输出关系为

$$Y(z) = \frac{1}{4}\mathcal{A}_0(z^2)\mathcal{A}_1(z^2) \tag{11.31}$$

表明这是一个无混叠的幅度保持正交镜像滤波器组。

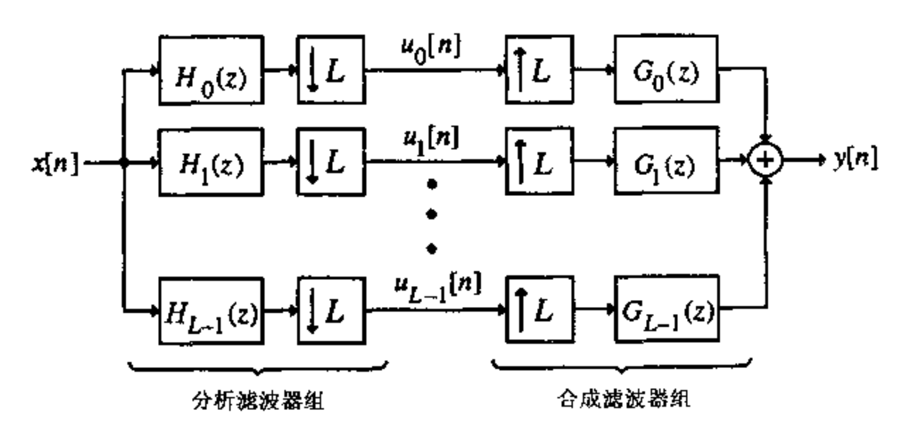


图 11.5 L 通道正交镜像滤波器组

考虑五阶巴特沃兹半带低通传输函数

$$H_0(z) = \frac{0.052\,786\,404\,5\,(1+z^{-1})^5}{1 + 0.633\,436\,854\,z^{-2} + 0.0557\,280\,9\,z^{-4}} \tag{11.32}$$

证明它可被分解成下面的形式:

$$H_0(z)=\frac{1}{2}\left[\left(\frac{0.105\,572\,81+z^{-2}}{1+0.105\,572\,81\,z^{-2}}\right)+z^{-1}\left(\frac{0.527\,864\,045+z^{-2}}{1+0.527\,864\,045\,z^{-2}}\right)\right] \tag{11.33}$$

由上述分解，可确定两个全通滤波器$\mathcal{A}_0(z)$和$\mathcal{A}_1(z)$。

编写一个 MATLAB 程序，用上面生成的两个全通滤波器仿真图 11.6 所示的双通道正交镜像滤波器组。接下来用 M 函数 `fir2` 产生一个具有三角幅度响应的、长度为 50 的序列，并用你已仿真的正交镜像滤波器组处理它。确定输出的幅度谱并证明它与输入具有相同的形式。全通滤波器的初始条件可能在输出幅度谱中引入一些明显的失真。怎样使它们带来的影响最小化？

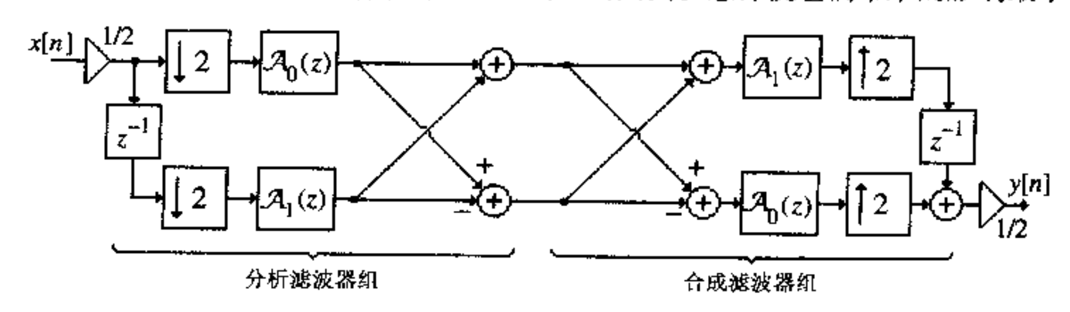


图 11.6　幅度保持双通道正交镜像滤波器组

若你的计算机有一个麦克风以及模数和数模转换板，可以考虑一下语音信号的获取及数字化，并用你已仿真的正交镜像滤波器组处理该信号，然后回放滤波器组的输出来观察其质量。

11.5　调制和解调

对于在信道上传输的低频信号，信号通过调制运算转换成高频信号。已调高频信号在接收端被解调且通过进一步处理提取所需的低频信号。在此我们将研究两类调制方案的运算：振幅调制和正交振幅调制。

例 11.10　振幅调制

在振幅调制中，高频正弦信号 $A\cos(\omega_o n)$ 称为载波信号，其振幅根据

$$y[n]=A\,x[n]\cos(\omega_o n) \tag{11.34}$$

被低频调制信号 $x[n]$改变，得到高频已调信号 $y[n]$。调制信号 $x[n]$的谱 $X(e^{j\omega})$的带度假定为 ω_m。因而，振幅调制可通过形成调制信号和载波信号的乘积来实现。$y[n]$的谱 $Y(e^{j\omega})$为

$$Y(e^{j\omega})=\frac{A}{2}X(e^{j(\omega-\omega_o)})+\frac{A}{2}X(e^{j(\omega+\omega_o)}) \tag{11.35}$$

假设 $\omega_o>\omega_m$，从 $y[n]$中得到 $x[n]$称为解调，它以两步进行。首先，将 $y[n]$和同载波具有相同频率的正弦信号相乘得到

$$r[n]=y[n]\cos(\omega_o n)=A\,x[n]\cos^2(\omega_o n)=\frac{A}{2}x[n]+\frac{A}{2}x[n]\cos(2\omega_o n) \tag{11.36}$$

乘积信号由被因子$\frac{1}{2}$尺度化的原调制信号 $x[n]$和载波频率为 $2\omega_o$的已调幅信号组成。将 $r[n]$通过截止频率为 ω_c，$\omega_m<\omega_c<2\omega_o-\omega_m$ 的低通滤波器滤波，可以得到原调制信号。滤波器的输出则是调制信号尺度化后的结果。

图 11.7 显示了幅度调制和解调方案的方框图。

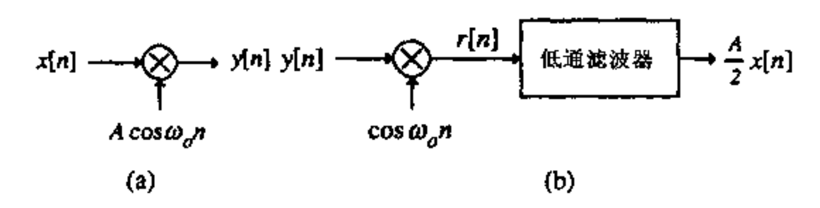


图 11.7 振幅调制和解调方案示意图:(a)调制器,(b)解调器

编写一个 MATLAB 程序来仿真振幅调制和解调方案的运算。

例 11.11 正交振幅调制

为了理解正交振幅调制(QAM)方法的基本思想,考虑两个带宽为 ω_m 的低频信号 $x_1[n]$ 和 $x_2[n]$。这两个调制信号分别被两个载波信号 $A\cos(\omega_o n)$ 和 $A\sin(\omega_o n)$ 单独地调制,并被求和,得到一个复合信号 $y[n]$:

$$y[n] = Ax_1[n]\cos(\omega_o n) + Ax_2[n]\sin(\omega_o n) \tag{11.37}$$

注意,两个载波信号有同一载波频率 ω_o,但有一个 90°的相位差别。通常,载波 $A\cos(\omega_o n)$ 称为同相成分,而载波 $A\sin(\omega_o n)$ 称为正交成分。复合信号 $y[n]$ 的谱 $Y(e^{j\omega})$ 此时为

$$\begin{aligned} Y(e^{j\omega}) = &\frac{A}{2}\left\{X_1(e^{j(\omega-\omega_o)}) + X_1(e^{j(\omega+\omega_o)})\right\} \\ &+ \frac{A}{2j}\left\{X_2(e^{j(\omega-\omega_o)}) - X_2(e^{j(\omega+\omega_o)})\right\} \end{aligned} \tag{11.38}$$

为了得到原始调制信号,复合信号分别乘以同相及正交成分得到两个信号:

$$r_1[n] = y[n]\cos(\omega_o n), \qquad r_2[n] = y[n]\sin(\omega_o n) \tag{11.39}$$

将式(11.37)中的 $y[n]$ 代入到式(11.39),通过运算后得到

$$\begin{aligned} r_1[n] &= \frac{A}{2}x_1[n] + \frac{A}{2}x_1[n]\cos(2\omega_o n) + \frac{A}{2}x_2[n]\sin(2\omega_o n) \\ r_2[n] &= \frac{A}{2}x_2[n] + \frac{A}{2}x_1[n]\sin(2\omega_o n) - \frac{A}{2}x_2[n]\cos(2\omega_o n) \end{aligned} \tag{11.40}$$

通过截止频率为 ω_m 的滤波器对 $r_1[n]$ 和 $r_2[n]$ 进行低通滤波得到两个调制信号。正交振幅调制和解调原理的框图如图 11.8 所示。

编写一个 MATLAB 程序来仿真正交振幅调制及解调的运算。

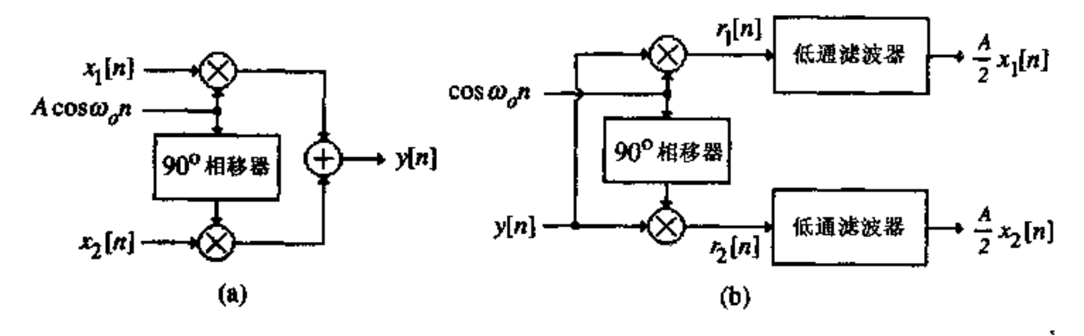


图 11.8 正交振幅调制和解调方案示意图:(a)调制器,(b)解调器

11.6 数字数据传输

二进制数据通常以脉冲串串行传输。然而,为了如实地提取传输的信息,接收器需要复均衡过程来补偿信道的非理想性并充分使用信道带宽。为了解决由于脉冲串数据传输带来的问题,可首选用于数字数据传输的多载波调制/解调方案。

例 11.12 离散多频声传输

多载波调制的一个广泛应用是*离散多频声传输*(DMT),其中调制和解调过程通过离散傅里叶变换实现,它可以用快速离散傅里叶变换来实现[Cio91],[Pel80]。为了理解离散多频声传输的基本思想,考虑抽样率为 F_T 的两个 $M-1$ 实数据序列$\{a_k[n]\}$和$\{b_k[n]\}$的传输,$1\leqslant k\leqslant M-1$。根据

$$\alpha_k[n]=\begin{cases}0, & k=0,\\ a_k[n]+j\,b_k[n], & 1\leqslant k\leqslant \frac{N}{2}-1\\ 0, & k=\frac{N}{2}\\ a_{N-k}[n]-j\,b_{N-k}[n], & \frac{N}{2}+1\leqslant k\leqslant N-1\end{cases} \tag{11.41}$$

定义一个长度为 $N=2M$ 的新复序列$\alpha_k[n]$。通过应用离散傅里叶逆变换,上面一组 N 个序列被转换成另一组N 个新信号$\{u_\ell[n]\}$,为

$$u_\ell[n]=\frac{1}{N}\sum_{k=0}^{N-1}\alpha_k[n]\,W_N^{-\ell k},\quad 0\leqslant \ell\leqslant N-1 \tag{11.42}$$

其中 $W_N=e^{-j2\pi/N}$。注意,产生复序列$\{\alpha_k[n]\}$的方法可以确保其离散傅里叶逆变换$\{u_\ell[n]\}$是实序列。这 N 个信号都经过N 倍上抽样并且在时间上相互交织,产生在一个抽样率为 $N\,F_T$ (它被假定等于2 F_c)的混合信号。复合信号通过数模转换器和模拟重构滤波器的转换,成为一个模拟信号 $x_a(t)$。模拟信号 $x_a(t)$接下来在信道上传输。

在接收端,接收到的模拟信号 $y_a(t)$穿过一个模拟反混叠滤波器,接着通过抽样和保持(S/H),然后再以速率 $N\,F_T=2\,F_c$ 通过模数转换成为数字信号$\{y[n]\}$。接收到的数字信号被一个包含 $N-1$ 个单位延时的延时链解交织,其输出接下来经过 N 倍下抽样得到一组信号$\{u_\ell[n]\}$。对这 N 个信号做离散傅里叶变换得到N 个信号$\{\beta_k[n]\}$

$$\beta_k[n]=\sum_{\ell=0}^{N-1}v_\ell[n]\,W_N^{\ell k},\quad 0\leqslant k\leqslant N-1 \tag{11.43}$$

图11.9所示是整个离散多频声传输系统。若我们假定信道的频率响应有一个平坦的通带,并且假定模拟重构及反混叠滤波器是理想低通滤波器,且忽略模数及数模转换器的非理想效果,可以假定 $y[n]=x[n]$。在这种情况下,发射端的离散多频声传输结构的交织电路是一个理想多路复用器,因此可以推出

$$\begin{aligned}v_k[n]&=u_{k-1}[n],\quad 1\leqslant k\leqslant N-2,\\ v_0[n]&=u_{N-1}[n]\end{aligned} \tag{11.44}$$

换句话说,

$$\begin{aligned} \beta_k[n] &= \alpha_{k-1}[n-1], \quad 1 \leqslant k \leqslant N-2 \\ \beta_0[n] &= \alpha_{N-1}[n] \end{aligned} \tag{11.45}$$

编写一个 MATLAB 程序来仿真离散多频声传输数据的传输。

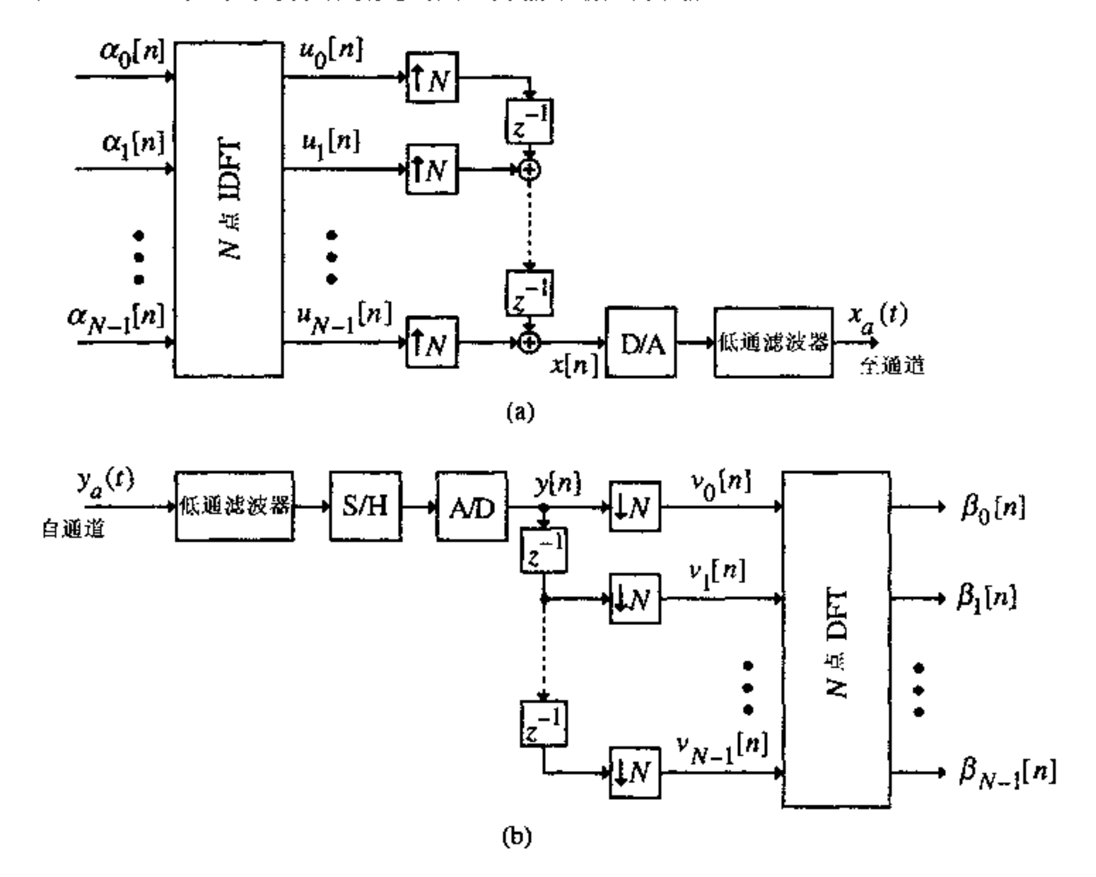


图 11.9 离散多频声传输方案:(a)传送器,(b)接收器

附录 A　MATLAB 简介

MATLAB 是一种用于科学计算且功能很强的高级程序设计语言。它易于学会，主要用来解决数值复杂的工程问题。本书所给出的习题是在假设读者并不精通 MATLAB 的情形下编写的。然而，这里包含 MATLAB 的一些基本概念，通过快速回顾它们，读者可很容易地理解程序并完成练习。更详细的内容可在[Mat94]，[Mat96]和[Mat97]中找到。

MATLAB 由内建于解释器中的函数或以 M 文件存在的函数组成，它是包含有实现某种算法的程序语句序列。一个全新的算法可以以只包含少量这些函数的程序写出，并可作为另一个 M 文件保存。

MATLAB 在电脑屏幕上以三种类型的窗口工作。它们是命令窗口、图形窗口和编辑器窗口。命令窗口的标题为 `Command`，图形窗口的标题为 `Figure No. 1`，而编辑器窗口的标题或者是一个已打开的现存 M 文件的名字，或者为 `Untitled`，代表创建一个新的 M 文件。命令窗口显示提示符 >>，表明它准备执行 MATLAB 命令。大部分输入命令的结果在命令窗口显示，同时这个窗口也可用来运行小程序和保存 M 文件。由画图命令产生的所有的图形在图形窗口出现。新 M 文件或旧 M 文件均从命令窗口运行，现有的 M 文件也可通过键入文件名从命令窗口运行。

在此附录的剩余部分中，我们将说明绝大多数常用函数，并复习一些与 MATLAB 有关的基本概念。

A.1　数字和数据表示

MATLAB 使用常规的十进制符号表示数字，负数可用一个位于最前面的负号来表示。MATLAB 中可表示的数字的近似范围从 10^{-308} 到 10^{308}。很大或很小的数字可以用指数形式表示。下面是一些有效数字表示的典型例子：

```
1234.56789     123456.789E-2     1.23456789e3     -1234.56789
```

注意，在指数前不应有空格。

MATLAB 中的数据以矩阵表示且不需要预先指定维数，元素可以是实数或复数。因此，一维离散时间信号可以表示为一个行向量或一个列向量。例如：

```
x=[3.5+4*j     -2.1-7.4*j     1.05-0.8*j     0     9.2*j];
```

该行向量表示一个长度为 5 的复数值信号 x，注意方括号表示 x 是一个矩阵。同样请注意，复数的虚部用算子 * 和字母 j 表示。虚部也可用字母 i 代替字母 j 来表示。如上所示，+ 或 - 号的两边应该无任何空格。矩阵的行中的元素也可用逗号分开，如下所示：

```
x=[3.5+4*j,     -2.1-7.4*j,     1.05-0.8*j,     0,     9.2*j];
```

方括号末端的分号表示数据输入后不会在命令窗口显示。若上述数据后没有分号，则 MATLAB 将在命令窗口显示

```
x =
   Columns 1 through 4
   3.5000 + 4.0000i    -2.1000 - 7.4000i    1.0500 - 0.8000i    0
   Column 5
   0 + 9.2000i
```

同样,若需要,x 的实际值可以通过在命令窗口键入 x 显示出来。

矩阵元素可以用两种不同的方式输入。既可以在用分号分开的单行键入,又可以在不同的行键入。例如,3×4 矩阵 A

$$A=\begin{bmatrix}1 & 3 & 5 & 7\\ 2 & 4 & 6 & 8\\ 9 & 11 & 13 & 15\end{bmatrix}$$

可以以

```
A=[1   3   5   7;   2   4   6   8;   9   11   13   15];
```

或以

```
A=[1   3   5   7
   2   4   6   8
   9   11   13   15];
```

的形式输入。

MATLAB 中向量的系数从 1 开始。例如,在上面的向量 x 中,x(1)是 3.5000 + 4.0000i,x(2)是 -2.1000 - 7.4000i,依次类推。类似地,矩阵 A 的第一行中的第一个元素为 A(1,1),第一行中的第二个元素为 A(1,2),依次类推。系数不能小于 1,或大于所考虑的向量或矩阵的维数。

MATLAB 工作区中,排列的大小可以通过函数 size 来确定。例如,键入 size(x)后,我们得到结果

```
ans =
   1   5
```

向量的长度也可使用函数 length 得到。例如,键入 length(x)后得到结果

```
ans =
   5
```

转置运算用运算符 .'实现。X 的转置的表达式为 X.'。若 X 是一个带有复数值元素的矩阵,则 X'代表 X 的复共轭转置,而若 X 是一个具有实数值元素的矩阵,则 X'表示 X 的转置。

MATLAB 内的数据向量和矩阵可以通过包括数字的字符集来标记,如 x,x1,X,X1,XY 等。注意,MATLAB 通常区分小写和大写字母。

例 A.1

设 X 表示通过键入输入的 3×4 实数值矩阵

```
X=[1  2  3  4;  5  6  7  8;  9  10  11  12];
```

接下来在命令窗口键入 X,得到

```
ans =
    1     2     3     4
    5     6     7     8
    9    10    11    12
```

而键入 X′则得到

```
ans =
    1     5     9
    2     6    10
    3     7    11
    4     8    12
```

接下来考虑输入为一个 2×3 复数值矩阵 Y 的情形：

```
Y=[1+2*i, 3-4*i, 5+6*i, 7-8*i, 9+10*i, 11-12*i];
```

键入 Y 产生

```
Y=
   1.0000 + 2.0000i    3.0000 - 4.0000i     5.0000 + 6.0000i
   7.0000 - 8.0000i    9.0000 + 10.0000i   11.0000 - 12.0000i
```

而键入 Y′得到

```
ans =
   1.0000 - 2.0000i      7.0000 + 8.0000i
   3.0000 + 4.0000i      9.0000 - 10.0000i
   5.0000 - 6.0000i     11.0000 + 12.0000i
```

键入 Y.′可得到 Y 的转置

```
ans =
   1.0000 + 2.0000i      7.0000 - 8.0000i
   3.0000 - 4.0000i      9.0000 + 10.0000i
   5.0000 + 6.0000i     11.0000 - 12.0000i
```

A.2　算术运算

如下所示，MATLAB 中有两种不同类型的算术运算用来对已存储的数据进行操作，其中 X 和 Y 表示两个不同的矩阵。若 X 和 Y 具有相同的维数，X 和 Y 的相加由表达式 X+Y 实现。相加运算 + 也可用来把一个标量加到一个矩阵中。同样地，从 X 减去 Y 由表达式 X-Y 实现。减法运算 - 也可用来将一个标量从一个矩阵中减去。

若 X 的列数与 Y 的行数相同，矩阵相乘 X*Y 产生 X 和 Y 的线性代数乘积。乘法运算 * 也可用来将一个标量乘以一个矩阵。若 X 和 Y 有同样的维数，X.*Y 是一个数组相乘，它形成 X 和 Y 的逐元素的乘积。

若 Y 是一个方阵，而 X 是一个具有与 Y 的列数相同的矩阵，则矩阵右除 X/Y 相当于 X*inv(Y)，其中 inv(Y)表示 Y 的逆。若它们中的一个是标量，右除运算 X/Y 也能实现。若 Y 是一个方阵而 X 是一个带有与 Y 的行数相同的矩阵，则矩阵左除 X\Y 相当于 inv(Y)*X。

若 X 和 Y 有相同的维数,数组右除用表达式 X./Y 实现,得到一个矩阵,其第(r,s)个元素由 X(r,s)/Y(r,s)给出。

若在一个语句中使用乘法运算,在求取这个表达式时遵循通常的优先规则。然而,括号可以改变运算的优先顺序。

矩阵的算术运算在下例中说明。

例 A.2

设 X=[1 2 3; 4 5 6]且 Y=[12 11 10; 9 8 7],则 X+Y 得到

```
ans =
    13    13    13
    13    13    13
```

而 X-Y 得到

```
ans =
    -11    -9    -7
    -5     -3    -1
```

运算 X+3 的结果为

```
ans =
    4     5     6
    7     8     9
```

而运算 X*3 的结果为

```
ans =
    3      6     9
    12     15    18
```

键入 X.*Y 生成

```
ans =
    12    22    30
    36    40    42
```

键入 X*Y'得到

```
ans =
    64    46
    163    118
```

而键入 X'*Y 得到

```
ans =
    48    43    38
    69    62    55
    90    81    72
```

考虑两个矩阵 X=[1 2 3; 4 5 6; 7 8 9]和 Y=[1 1 2; 2 2 3; 1 3 4]。
则 X/Y 生成

```
ans =
     0.5000        0   0.5000
     2.5000   3.0000   0.5000
    -5.5000   6.0000   0.5000
```

而 Y \ X 得到

```
ans =
     0    0    0
     5    4    3
    -2   -1    0
```

A.3　关系运算符

在 MATLAB 中，关系运算符 <，<=，>，>=，== 和 ~=，分别表示比较运算小于、小于等于(≤)、大于、大于等于(≥)、等于和不等于(≠)。使用这些运算符对两个同样大小的矩阵进行比较时，MATLAB 将把对应的元素逐个进行比较，其结果也将以同样大小的矩阵出现，且当关系为真时，矩阵中的该元素被设为 1，而当关系为假时，该元素被设为 0。在复数值的矩阵情况下，运算符 <，<=，> 和 >= 将只用来比较矩阵元素的实数部分，但运算符 == 和 ~= 用来比较实数和虚数两部分。

下例将说明这些运算符的用法。

例 A.3

考虑两个矩阵 C = [1 2 3; 4 5 6] 和 D = [1 7 2; 6 5 1]。则在 C 和 D 上应用上述关系运算符的结果如下：

```
C < D = 0   1   0
        1   0   0

C > D = 0   0   1
        0   0   1

C <= D = 1   1   0
         1   1   0

C >= D = 1   0   1
         0   1   1

C == D = 1   0   0
         0   1   0

C ~= D = 0   1   1
         1   0   1
```

A.4　逻辑运算符

在 MATLAB 中，三个逻辑运算 &，| 和 ~ 分别实现与(AND)、或(OR)及非(NOT)运算。当应

用到矩阵时,它们将逐个元素进行运算,用0表示FALSE而用1表示TRUE。下例将举例说明这些运算符的用法。

例 A.4

考虑两个矩阵A = [1101]和B = [0100]。在A和B上应用上述逻辑运算符的结果如下:

```
A & B = 0    1    0    0
A | B = 1    1    0    1
   ~A = 0    0    1    0
```

A.5 流程控制

MALTAB的流程控制命令是break,else,elseif,end,error,for,if,return和while。这些命令允许有条件地执行某些程序指令。命令for用于以给定的次数重复一组程序指令。命令if用来有条件地执行一组程序指令,而命令while可以用来以一个无法确定的次数重复程序指令。跟在命令for,while和if后面的指令必须用命令end结束。命令break用于结束一个循环的执行。命令else和elseif需与命令if一起使用,它可在一个循环的执行中使其有条件地停止。命令error用于显示错误消息并中断函数。

这些命令的用法将在下例中说明。

例 A.5

考虑生成一个以给定的第一个元素x(1)开始的长度为N的整数序列,且随后的每一个元素都以给定的正整数D线性增加。产生并显示这个序列的MATLAB程序如下:

```
N = 10;
D = 3;
x = [5 zeros(1,N - 1)];
for k = 2:N
    x(k) = x(k - 1) + D;
end
disp('产生的序列为');disp(x)
```

例 A.6

考虑生成一个以给定的第一个元素x(1)开始的长度为N的整数序列,且随后的每一个元素都以给定的正整数D线性增加,直到一个元素等于R * D + x(1)为止,这里R是一个正整数,而随后的每一个元素又以数量D线性减少,直到这个元素等于x(1),然后重复这个过程。产生并显示这个序列的MATLAB程序给出如下:

```
N = 15; D = 3;
x = [5 zeros(1,N - 1)];
for k = 2:N
    x(k) = x(k - 1) + D;
    if x(k) = = 3 * D + x(1)
      D = - abs(D);
```

```
        elseif x(k) = = x(1)
          D = abs(D);
        end
    end
    disp('产生的序列为');disp(x)
```

例 A.7

下面的程序说明了命令 break 的用法。程序得到以一个给定初值 y 开始的一系列数字的和,后面的每一个数以固定的正增量 D 增加;当总和超过 100 时,即停止加法过程;接下来显示总和。

```
    y = 5; D = 3;
    while 1
        y = y + D;
        if y > 100,break,end
    end
    disp('y 是');disp(y)
```

A.6 特殊字符和变量

MATLAB 使用许多特殊字符和字表示特定的项。这些字符和字不应该再用于任何其他用途。例如,若将字母 i 或字母 j 作为变量使用,它就不能用来表示复数的虚部。因此,一种良好的习惯做法是限定字母 i 或字母 j 专门用做复数的虚数部分的表示。

有几个永久变量,用户不能清除它们的值,且不能将它们用来表示任何其他量。字 pi 用来表示 π,因而 sin(pi/4)的结果为 0.707 106 781 186 55,它等于 $1/\sqrt{2}$。变量 eps 等于 2^{-52} 并且是一个对于某个确定计算(例如一个矩阵的秩)的精度的容限。它可由用户设为任何其他值。NaN 表示非数字,在计算未定义的运算(例如 0/0 和 $\infty-\infty$)时得到它。inf 表示 $+\infty$,由诸如被零除这样的运算(如 2/0)或溢出(如 e^{1000})得到。变量 ans 存储最近运算的结果。

方括号[]用来输入矩阵和向量。矩阵的元素可用空格或逗号分开,分号;表示一个矩阵中一行的结束,它也可用来屏蔽显示。算术表达式中的优先可用圆括号()来表明,圆括号也用于包含一个排列的序号和函数的自变量。排列的转置的运算符是',然而两个这种符号可用来表示一个引用。例如,'dsp program'是一个在在引号中包含字符的 ASCII 码的向量。跟在百分数符号%后的任何文字都表示注释,而不视为程序指令。

在 MATLAB 中,冒号:有很多不同的应用。它用于产生向量、下标矩阵以及重复执行一组指令。例如,若 M < N,x = M:N 产生向量

x = [M   M+1   M+2   …   N]

然而,若 M > N,x = M:N 是一个空矩阵,由[]表示。命令 x = M:k:N 产生向量

x = [M   M+k   M+2k   …   N]

其中 k 可以是正或负整数。注意,若 k > 0 及 M > N,或 k < 0 及 M < N,x = M:k:N 产生空矩阵[]。

冒号也用于选择一个矩阵或一个向量特定的行、列和元素。例如,Y(:,N)表示 Y 的第 N 列。同样,Y 的第 M 行由 Y(M,:)表示。Y(:,M:N)等价于 Y(:,M),Y(:,M+1),…,Y(:,N)。最后,Y(:)等价于由 Y 的所有列连接而形成的一个列向量。

A.7 输出数据格式

MATLAB 中的所有算术运算都以双精度进行。然而,在命令窗口中可用不同的格式来显示操作的结果。若所有的结果是严格的整数,则它们以无小数点的形式显示。若一个或更多数据元素不是整数,则结果可按不同的精度显示。format short 显示五个有效小数位,这是默认的格式。format short e 显示带有两个正或负十进制指数的五个有效小数位。format long 显示 15 个有效小数位,而 format long e 显示 15 个有效小数位加两个正或负十进制指数。显示结果还有三种其他的格式,但这三种格式在信号处理应用中不常用。

A.8 图形

MATLAB 能将计算结果以高级的图形方式显示出来。在许多情况下,我们仅涉及二维图形,只在一些特殊情况下用到三维图形。对于二维图形,图形可以以不同的形式表示——坐标轴既可用线性尺度,又可用对数尺度,或者一个轴用线性尺度而另一个轴用对数尺度。我们可以在图中加入栅格线,在两个轴上加上标记,在图的顶部加入标题,等等。文本既可以通过鼠标放置在图形的任何位置,又可以用程序中的命令指定其起始位置。此外,通过在画图命令中加入适当的参数,可以在图形中显示特定的线型、图像符号和颜色。

对于三维数据,图形也可以以不同的形式表示——对图形的一个、两个或三个坐标轴,既可以用线性尺度,也可用对数尺度。例如,线和点可以在三维尺度中画出。也可产生轮廓图、三维透视图、平面图、伪彩色图等。

下一节中的 M 文件将说明几个图形命令的用法。

A.9 M 文件:脚本及函数

M 文件是用文字处理器或文本编辑器生成的一个 MATLAB 语句序列,并且文件名必须以*文件名* .m 的形式保存。M 文件名需要以字母开头且至多不超过 18 个字母和/或数字(或下划线)。然而,有些字符,例如连字符 - 和小数点 .,不能用于文件名。同样,不能用已经存在的 M 文件名来命名一个新 M 文件。一个 M 文件可引用其他现有的 M 文件名。

如同 ASCII 文本文件一样,新程序的每一条语句可以在编辑器窗口中逐行键入,并且可以用文本编辑器或文字处理器进行编辑①。整个程序完成后可作为一个 M 文件保存。

有两类 M 文件:*脚本*和*函数*。函数文件必须在所有程序语句的第一行包含单词 function。函数文件中的自变量可以由其他的 M 文件中传递而来。在函数文件内部用到的变量都是局部变量。

---

① 注意,UNIX 工作站无内置文本编辑器。因此,程序必须用其他类型的编辑器编写。

脚本文件完全使用工作区数据。函数文件的第一行包含单词 function，它并不完全使用工作区数据。在函数文件内定义的变量是局部操作的，只有函数文件的自变量可能被传递。当一个函数执行完毕，所有的局部变量就丢失，而只保留特定的待传递的数值。

以下给出一个函数文件 runsum，以作为简单的例子。

```
function y = runsum(x)
%计算一个向量 x 的均值
L = length(x);
y = sum(x)/L;
```

脚本文件 lowpass.m 的简单例子如下：

```
%脚本 M 文件 lowpass.m
%程序实现低通滤波
%对一个随机信号使用三点平均
%程序使用函数文件 runsum
z = zeros(1,11);data = randn(size(z));
u = [zeros(1,3) data];
N = 3; %N 是滤波器长度
for k = 1:10;
    w = u(k:k + N);
    z(k) = runsum(w);
end
n = 0:10;
%用实线画出噪声并
%用虚线画出平滑后的形式
plot(n,data,'r - ',n,z,'b--');grid
xlabel('时间序号 n');
ylabel('振幅');
gtext('噪声数据');gtext('平滑后的数据');
```

图 A.1 显示了执行文件 lowpass.m 后产生的图形。

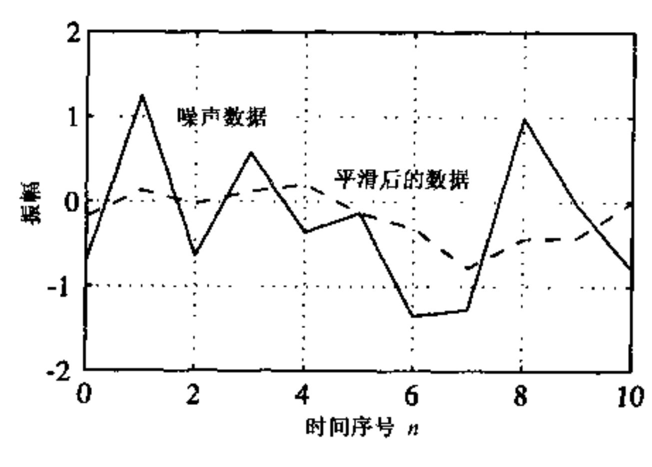


图 A.1　信号平滑举例

注意，函数文件 runsum 使用内置函数 sum。同样，脚本文件 lowpass.m 使用函数文件 runsum。

A.10　MAT 文件

为方便以后使用,由一个 MATLAB 程序生成的数据可以以二进制文件的形式保存,这种文件称为 MAT 文件。例如,通过执行前一节的程序 lowpass.m 而产生的噪声数据,可用命令

```
save noise.mat datc
```

保存。以后它可用命令

```
load noise
```

取回,以便用于其他的 MATLAB 进程。

数据也可用 ASCII 形式保存。例如,在程序 lowpass.m 完成后,我们可用命令

```
result = [noise; z];
```

产生一个包含噪声数据和平滑数据的 2×11 矩阵,接下来用命令

```
save tempo.dat result -ascii
```

以 ASCII 形式保存矩阵 result。此后,可以用命令

```
load tempo
```

取出存储的数据。

A.11　显示

可以用命令 print 生成当前图形窗口的副本。该命令有许多形式,见 *MATLAB Reference Guide*[Mat94]。在 PC 或 Mac 环境中,一幅图形也可先复制到剪贴板中,然后再插入到文字处理文档。这种方法能产生较小尺寸的图形,并且可以将图形粘贴到文本中。

A.12　诊断和帮助工具

MATLAB 有很好的诊断能力,这使得纠正执行中发现的任何错误变得非常容易。若任何程序语句有错误,程序将停止执行并在命令窗口显示一目了然的错误信息。例如,输入实数 1.23456789e3 时在指数前加了一个空格,MATLAB 将给出错误信息

```
???  1.23456789  e3
                 |
Missing operator,comma,or semi-colon.
```

若用冒号代替了小数点,将实数 1.23456789e3 输入为 1:23456789e3,则会引起错误信息

```
???  Error using ==> colon
Maximum variable size allowed by the program is exceeded.
```

MATLAB 通过命令 help 对大多数主题提供在线帮助信息。若在命令窗口中不带自变量直接

键入 help,在窗口将显示包含 MATLAB 文件的目录清单。若想获得特定的 M 文件或目录的帮助,可将文件或目录名作为一个自变量。例如键入 help runsum 得到

```
Computes the mean of a vector x
```

同样,键入 help lowpass 出现

```
A Script M - file to Perform Lowpass Filtering
Using Three - Point Averaging
    Program uses the function file runsum
```

键入 who 可得一份在工作区的变量的清单。要得到关于变量大小的信息,可使用命令 whos。其他有用的命令是 what,which,lookfor,echo 和 pause。

命令 what 列出当前目录中的所有文件,而命令 what dryname 则列出在 MATLAB 的搜索路径中名为 dryname 的目录中的文件。命令 which 用于查找在 MATLAB 的搜索路径上的函数和文件。命令 lookfor abc 从头到尾搜寻在 MATLAB 的搜索路径上的所有帮助条目,并在第一条注释线中寻找字符串 abc。命令 echo 用来调试一个新程序,并列出一个程序执行期间被调用的所有 M 文件。这个命令有几种形式,详情请参阅 *MATLAB Reference Guide* [Mat94]。命令 pause 在指定的点临时停止程序的执行;按键盘上的任意键后,可在该点继续程序的执行。当程序产生大量图形时,这个命令特别有用,而且若命令 pause 插入在每个绘图命令之后,则每一个图形可被单独地检查或复制。

A.13 注释

即使 MATLAB 使用双倍精度算术运算,在计算中使用的数字近似值也可能会在结果中产生误差。必须考虑这种情况来恰当地解释结果。例如,当结果应理想地等于 0 时,表达式 1 - 0.1 - 0.3 - 0.2 - 0.2 - 0.1 - 0.1 以 format long 形式得到的输出为 5.551115123125783e - 17。另一方面,将该表达式稍微修改为 1 - (0.1 + 0.3 + 0.2 + 0.2 + 0.1 + 0.1)后,可得出正确的结果 0。

附录 B　用到的 MATLAB 命令总结

下面对本书中用到的 MATLAB 函数做一个简单回顾。关于这些函数的详细内容,请使用 help 命令查阅。

函数	**描述**
abs	计算绝对值
angle	计算相角(单位为弧度)
axis	手工设置图中坐标的尺度
blackman	产生布莱克曼窗系数
break	终止循环的执行
butter	设计所有四种类型数字或模拟巴特沃兹滤波器
buttord	选择数字或模拟巴特沃兹传输函数的最小阶数
ceil	朝 + ∞ 方向最接近整数进行取整
cheb1ord	选择模拟或数字切比雪夫 1 型滤波器的最小阶数
cheb2ord	选择模拟或数字切比雪夫 2 型滤波器的最小阶数
chebwin	产生多尔夫-切比雪夫窗系数
cheby1	设计所有四种类型的数字或模拟切比雪夫 1 型滤波器
cheby2	设计所有四种类型的数字或模拟切比雪夫 2 型滤波器
clf	从当前图形删除所有对像
conj	计算复共轭
conv	两个多项式相乘
cos	计算余弦
decimate	以整数因子降低序列的抽样率
deconv	进行多项式除法
disp	在屏幕上显示矩阵或文本
echo	在执行中禁止(或允许)在屏幕上显示 M 文件
ellip	设计所有四种类型的数字或模拟椭圆滤波器
ellipord	选择数字或模拟椭圆传输函数的最小阶数
else	在一个 if 循环中描述另一语句块
elseif	条件执行在一个 if 循环内的语句块
end	终止一个循环
eps	表示浮点相对精度
error	显示一个错误消息
exp	计算指数
fft	计算离散傅里叶变换系数
filter	以转置级联 II 型结构实现的一个 IIR 或 FIR 滤波器的滤波器数据
filtfilt	进行数据的零相位滤波
fir1	用加窗傅里叶级数法设计所有四类线性相位 FIR 滤波器

fir2	用加窗傅里叶级数法设计具有任意幅度响应的线性相位 FIR 滤波器
fix	朝零方向取整
fliplr	将矩阵进行左右方向翻转
flops	计算浮点运算累积数
for	以一个给定数目的次数重复执行语句块
format	控制输出显示的格式
freqs	在指定频率点计算一个模拟传输函数的复频率响应
freqz	在指定频率点计算一个数字传输函数的复频率响应
function	产生新的 M 函数
grid	在当前图形上增加或减少网格线
grpdelay	在指定频率点计算一个数字传输函数的群延迟
gtext	在鼠标的帮助下于图形上放置文本
hamming	产生海明窗系数
hanning	产生汉宁窗系数
help	对 MATLAB 函数和 M 文件提供在线帮助
hold	保持当前图形
if	条件执行语句
ifft	计算离散傅里叶逆变换的系数
imag	确定一个复数或矩阵的虚部
impz	计算给定数目的一个数字传输函数的冲激响应系数
input	要求由用户提供的数据
interp	以一个整数因子增加一个序列的抽样率
inv	矩阵求逆
kaiser	确定凯泽窗系数
kaiserord	确定凯泽窗的滤波器阶数和参数?
latc2tf	由 Gray-Markel 实现给定的格型参数和前馈系数确定传输函数
legend	用一个给定字符串作为标记在当前图上插入说明
length	确定向量的长度
linspace	产生线性间隔的向量
load	从磁盘文件中取回保存好的数据
log10	计算常用对数
lookfor	通过所有 help 条目提供关键字搜索
max	确定向量的最大元素
min	确定向量的最小元素
NaN	不是一个数
nargin	表明函数 M 文件体内变量的数目
num2str	将一个数字转换成它的字符串表示
ones	产生所有元素是 1 的向量或矩阵
pause	暂时停止执行直到用户按任何键
pi	返回最接近 π 的浮点数
plot	产生线性二维图形
poly2rc	确定一个 IIR 全通传输函数的级联实现的系数

rand	产生在区间(0,1)间均匀分布的随机数和矩阵
randn	产生具有零均值和单位方差的正态分布的随机数和矩阵
real	确定一个复数或矩阵的实部
rem	确定矩阵除以同样大小的矩阵后的余数
remez	用 Parks-McCellan 算法设计线性相位 FIR 滤波器
remezord	和 remez 命令一起使用确定近似阶数、归一化频带边界、频带幅度级和权值
resample	以某个分数改变一个序列的抽样率
residue	确定以 z 的多项式的比表示的离散时间传输函数的部分分式
residuez	确定以 z^{-1} 的多项式的比表示的离散时间传输函数的部分分式
return	引起返回键盘或调用函数
roots	确定多项式的根
save	在磁盘上保存工作区变量
sawtooth	产生周期为 2π 的锯齿波
sign	执行符号函数
sin	确定正弦
sinc	计算一个向量或排列的 sinc 函数
size	返回矩阵维数
sqrt	计算平方根
square	产生周期为 2π 的方波
stairs	画阶梯图
stem	数据序列的图形表示
subplot	将图形窗口分成多个矩形窗格来显示多个图形
sum	确定一个向量的所有元素的和
tf2latc	确定一个 IIR 传输函数的 Gray-Markel 实现的格型参数和前馈系数
tf2zp	确定给定传输函数的零点、极点和增益
title	给当前图像加上标题文本
unwrap	消除相位角的跳变来在分支切口上提供平滑的过渡
what	提供文件的目录清单
which	定位函数和文件
while	以不确定的次数重复语句
who	列出内存中的当前变量
whos	列出内存中的当前变量,以及变量大小、是否有非零虚部
xlabel	在当前二维图形中的 x 轴写指定文本
ylabel	在当前二维图形中的 y 轴写指定文本
zeros	生成元素是零的一个向量或矩阵
zp2sos	由一个给定的零－极－增益表示确定一个等效二阶表示
zp2tf	由一个传输函数的给定的零点、极点和增益确定其分子和分母系数
zplane	在 z 平面中显示极点和零点

参考文献

[Abr72] M. Abramowitz and I.A. Segun, editors. *Handbook of Mathematical Functions*. Dover Publications, New York NY, 1972.

[Bag98] S. Bagchi and S.K. Mitra. *Nonuniform Discrete Fourier Transform and Its Signal Processing Applications*. Kluwer, Boston MA, 1998.

[Bat80] M.R. Bateman and B. Liu. An approach to programmable CTD filters using coefficients 0, +1, and −1. *IEEE Trans. on Circuits and Systems*, CAS-27:451-456, June 1980.

[Cio91] J. Ciofi. *A Multicarrier Primer*. ANSI T1E1.4 Committee Contribution, Boca Raton FL, November 1991.

[Con70] A.C. Constantinides. Spectral transformations for digital filters. *Proc. IEE (London)*, 117:1585–1590, August 1970.

[Cro75] R.E. Crochiere and A.V. Oppenheim. Analysis of linear digital networks. *Proc. IEEE*, 62:581–595, April 1975.

[Gas85] L. Gaszi. Explicit formulas for lattice wave digital filters. *IEEE Trans. on Circuits & Systems*. CAS-32:68–88, January 1985.

[Gra73] A.H. Gray, Jr. and J. D. Markel. Digital lattice and ladder filter synthesis. *IEEE Trans. on Audio and Electroacoustics*. AU-21:491–500, December 1973.

[Her73] O. Herrman, L.R. Rabiner, and D.S.K. Chan. Practical design rules for optimum finite impulse response lowpass digital filters. *Bell System Tech. J.*. 52:769-799, 1973.

[Jac70] L.B. Jackson. On the interaction of roundoff noise and dynamic range in digital filters. *Bell System Technical Journal*, 49:159–184, February 1970.

[Jac96] L.B. Jackson. *Digital Filters and Signal Processing*. Kluwer, Boston MA, third edition, 1996.

[Jar88] P. Jarske, Y. Neuvo, and S.K. Mitra. A simple approach to the design of FIR filters with variable characteristics. *Signal Processing*, 14:313–326, 1988.

[Kai74] J.F. Kaiser. Nonrecursive digital filter design using the I_0-sinh window function. *Proc. 1974 IEEE International Symposium on Circuits and Systems*. pages 20-23, San Francisco CA, April 1974.

[Kra94] T.P. Krauss, L. Shure and J.N. Little. *Signal Processing TOOLBOX for use with MATLAB*. The Mathworks, Inc., Natick MA, 1994.

[Lim86] Y.C. Lim. Frequency-response masking approach for the synthesis of sharp linear phase digital filters. *IEEE Trans. on Circuits and Systems*. CAS-33:357-364, April 1986.

[Mar92] A. Mar, editor. *Digital Signal Processing Applications Using the ADSP-2100 Family*. Prentice-Hall, Englewood Cliffs NJ, 1992.

[Mat94] *MATLAB Reference Guide*. The Mathworks, Inc., Natick MA, 1994.

[Mat96] *MATLAB Signal Processing Toolbox User's Guide*. The Mathworks, Inc., Natick MA, 1996.

[Mat97] *Using MATLAB, Version 5*. The Mathworks, Inc., Natick MA, 1997.

[Mit74a] S.K. Mitra and K. Hirano. Digital allpass networks. *IEEE Trans. on Circuits and Systems*, CAS-21:688–700, 1974.

[Mit74b] S.K. Mitra, K. Hirano, and H. Sakaguchi. A simple method of computing the input quantization and the multiplication round-off errors in digital filters. *IEEE Trans. on Acoustics, Speech, and Signal Processing*. ASSP-22:326–329, October 1974.

[Mit77a] S.K. Mitra, K. Mondal, and J. Szczupak. An alternate parallel realization of a digital transfer function. Proc. IEEE (Letters), 65:577–578, April 1977.

[Mit77b] S.K. Mitra and C.S. Burrus. A simple efficient method for the analysis of structures of digital and analog systems. *Archiv für Elektrotechnik und Übertragungstechnik*, 31:33-36, 1977.

[Mit90] S.K. Mitra, Y. Neuvo, and H. Roivainen. Design and implementation of digital filters with variable characteristics. *International Journal on Circuit Theory and Applications*, 18:107-119, 1990.

[Mit98a] S. K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, New York NY, 1998.

[Mit98b] S.K. Mitra and A. Makur. Warped discrete Fourier transform. *Proc. IEEE Workshop on Digital Signal Processing*, Bryce UT, August 1998.

[Neu84] Y. Neuvo, C-Y Dong, and S.K. Mitra. Interpolated finite impulse response filters. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-32:563-570, June 1984.

[Neu87] Y. Neuvo, G. Rajan, and S.K. Mitra. Design of narrow-band FIR bandpass digital filters with reduced arithmetic complexity. *IEEE Trans. on Circuits and Systems*, CAS-34:409-419, April 1987.

[Par72] T. W. Parks and J. H. McClellan. Chebyshev approximation for nonrecursive digital filters with linear phase. *IEEE Trans. on Circuit Theory*. CT-19:189-194, 1972.

[Pel80] A. Peled and A. Ruiz. Frequency domain data transmission using reduced computational complexity algorithm. *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Denver, CO, pages 964-967, August 1980.

[She95] O. Shentov, S.K. Mitra, A.N. Hossen, and U. Heute. Subband DFT - Part I: Definition, interpretation and extension. *Signal Processing*, 41:261-277, No. 3, February 1995.

[Vai86] P.P. Vaidyanathan, S.K. Mitra, and Y. Neuvo. A new approach to the realization of low sensitivity IIR digital filters. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-34:350–361, April 1986.

[Vai87] P.P. Vaidyanathan and S.K. Mitra. A unified structural interpretation and tutorial review of stability test procedures for linear systems. *Proc. IEEE*, 75:478–497, April 1987.

[Vet88] M. Vetterli. Running FIR and IIR filtering using multirate techniques. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 36:730-738, May 1988.

索　　引